IoT Domain Analyst

Lab Record — Lab 1

- 8 February 2021

Programme	:	B.Tech(CSE)	Semester	:	Winter 2020–21
Course Title	:	IoT Domain Analyst – Lab	Code	:	ECE3502
			Slot	:	L5+L6
Name	:	**Aadhitya Swarnesh**	Registration. No	:	[redacted]
Faculty (s)	:	[redacted]	Expt. No	:	**1**

Experiment 1 :

To calculate the distance between two sets of numbers using

Euclidean distance.

Aim :

To calculate the Euclidean distance between numbers which are placed in a pair of arrays.

Description :

The Euclidean distance is calculated as follows :

$$d\left(p,q\right)=\sqrt{\sum_{i=1}^{n}\left(q_i-p_i\right)^2}$$

We use this formula to calculate the Euclidean distance between two arrays denoted in the formula as **p** and **q** respectively, and the summation runs through all the elements of the arrays.

Code :

The code that is used for this Experiment is as follows :

```
def euclidean(a, b) :
    summ = 0
    n = len(a)

    for i in range(n) :
        summ += (b[i]-a[i])**2

    summ = summ ** (1/2)

    return summ


a = [0, 3, 4, 5]
b = [7, 6, 3, -1]
print(euclidean(a, b))
```

Result :

The result for the obtained code displays the Euclidean distance between these two arrays, and the output for the same is as follows :

```
(base) Aadhityas-MacBook-Air:lab 1 aadhitya$ python euclidean.py
9.746794344808963
(base) Aadhityas-MacBook-Air:lab 1 aadhitya$
```

Experiment 2 :

To calculate the distance between two sets of numbers using

Manhattan distance.

Aim :

To calculate the Manhattan distance between numbers which are placed in a pair of arrays.

Description :

The Manhattan distance is calculated as follows :

$$distance = |\sum_{i}(x_i - y_i)|$$

We use this formula to calculate the Manhattan distance between two arrays denoted in the formula as **x** and **y** respectively, and the summation runs through all the elements of the arrays.

Here we take the absolute magnitude of the distance between the pair of numbers, which gives the Manhattan distance between them.

Code :

The code that is used for this Experiment is as follows :

```
def manhattan(a, b) :
    summ = 0
    n = len(a)

    for i in range(n) :
        summ += abs(b[i]-a[i])

    return summ


a = [20, 10, 30]
b = [10, 10, 20]
print(manhattan(a, b))
```

Result :

The result for the obtained code displays the Manhattan distance between these two arrays, and the output for the same is as follows :

```
(base) Aadhityas-MacBook-Air:lab 1 aadhitya$ python manhattan.py
20
(base) Aadhityas-MacBook-Air:lab 1 aadhitya$
```

Experiment 3 :

To calculate the distance between two sets of numbers using

Minkowski distance.

Aim :

To calculate the Minkowski distance between numbers which are placed in a pair of arrays, and using the given **p** value.

Description :

The Minkowski distance is calculated as follows :

$$d\left(a,b\right) = \sqrt[p]{\sum_{i=1}^{n}\left(a_i - b_i\right)^p}$$

We use this formula to calculate the Minkowski distance between two arrays denoted in the formula as **a** and **b** respectively, and the summation runs through all the elements of the arrays.

Here we take the **p**'th root of the distance between the difference of the pair of numbers raised to the **p**'th power, which gives the Minkowski distance between them.

Code :

The code that is used for this Experiment is as follows :

```
def minkowski(a, b) :
    summ = 0
    n = len(a)

    for i in range(n) :
        summ += (b[i]-a[i])**p

    summ = summ ** (1/p)

    return summ


a = [0, 3, 4, 5]
b = [7, 6, 3, -1]
p=3
print(minkowski(a, b, p))
```

Result :

The result for the obtained code displays the Minkowski distance between these two arrays, and the output for the same is as follows :

```
(base) Aadhityas-MacBook-Air:lab 1 aadhitya$ python minkowski.py
5.348481241239363
(base) Aadhityas-MacBook-Air:lab 1 aadhitya$
```