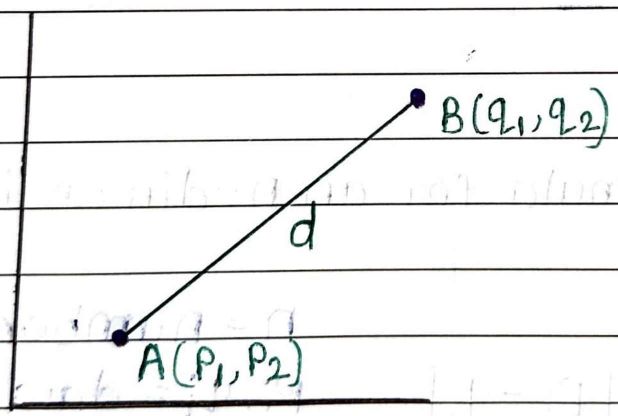
Understanding Distance Metrics used in Machine Learning

Culstering is an important part of data cleaning, used in the field of artificial intelligence, deep learning and data science. Distance metrics basically deal with finding the proximity or distance between data points and determining if they can be cluster together.

Commonly used distance metric-

Euclidean Distance or L2 Distance-

Euclidean Distance represent the shortest distance between two vectors.
It is the square root of sum of squares of differences between correspondind elements.



Formula for 2-Dimension of Euclidean Distance

$$d = \sqrt{(P_1 - q_1)^2 + (P_2 - q_2)^2}$$

Generalized formula for an n-dimensional space:-

$$D_e = \left[\sum_{i=1}^{n} (P_i - q_i)^2 \right]^{1/2}$$

n = number of dimensions
P_i, q_i = data points

Euclidean Distance or L2 Distance

[ ]

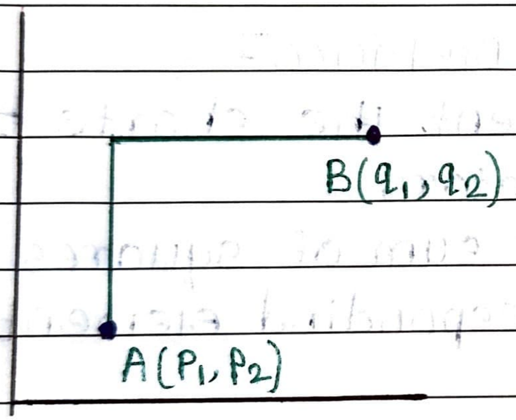
```python
#importing the library
from scipy.spatial import distance

#Defining the points

point_1=(4,5,6)
point_2=(8,9,10)

#computing the euclidean distance

Euclidean_distance = distance.euclidean(point_1,point_2)
print("Euclidean distance between point_1 and point_2 is:",Euclidean_distance)
```

```
Euclidean distance between point_1 and point_2 is: 6.928203230275509
```

Manhattan Distance or city block distance or L1 Distance

Manhattan Distance is the sum of absolute difference between points across all the dimensions.

To calculate Manhattan Distance, we will take sum of absolute distance in both the x and y directions.



Representation is 2-D

Formula for Manhattan Distance

$$d = |p_1 - q_1| + |p_2 - q_2|$$

Generalized formula for an n-dimensional space-

$$D_m = \sum_{i=1}^{n} |p_i - q_i|$$

n = number of dimensions

p_i, q_i = data points

Note - At N=1

Euclidean Distance = Manhattan Distance

Manhattan Distance or city block distance

```python
#importing the library
from scipy.spatial import distance

#Defining the points

point_1=(4,5,6)
point_2=(8,9,10)

#computing the manhattan distance

Manhattan_distance=distance.cityblock(point_1,point_2)
print("Manhattan distance between point_1 and point_2 is:",Manhattan_distance)
```

```
Manhattan distance between point_1 and point_2 is: 12
```

Minkowski Distance -

Minkowski Distance is the generalized form of Euclidean and Manhattan Distance

Formula for Minkowski Distance.

$$d = \left[\sum_{i=1}^{n} |p_i - q_i|^p \right]^{1/p}$$

p represent the order of the norm.

Note -

At $p=1$

Minkowski Distance = Manhattan / L_1 Distance

At $p=2$

Minkowski Distance = Euclidean / L_2 Distance

At $p=\infty$

Minkowski Distance = chebyshev / L_∞ Distance

Maximum value contribute to chebyshev Distance

$$\max\left(\underbrace{|p_1 - q_1|}_{dx_1}, \underbrace{|p_2 - q_2|}_{dx_2} \right)$$

e.g. $dx_1 = 10$ & $dx_2 = 5$

$10^\infty >>> 5^\infty$

Minkowski Distance

[1]

```
#Let’s calculate the Minkowski Distance formula of order 3

#importing the library
from scipy.spatial import distance

#Defining the points

point_1=(4,5,6)
point_2=(8,9,10)

#computing the minknowski distance

minkowski_distance = distance.minkowski(point_1, point_2, p=3)
print("Minknowski distance between point_1 and point_2 is:",minkowski_distance)
```

```
Minknowski distance between point_1 and point_2 is: 5.768998281229633
```

```python
#Let's calculate the Minkowski Distance formula of order 1

#importing the library
from scipy.spatial import distance

#Defining the points

point_1=(4,5,6)
point_2=(8,9,10)

#computing the minknowski distance

minkowski_distance = distance.minkowski(point_1, point_2, p=1)
print("Minknowski distance between point_1 and point_2 is:",minkowski_distance)

#computing the manhattan distance

Manhattan_distance=distance.cityblock(point_1,point_2)
print("Manhattan distance between point_1 and point_2 is:",Manhattan_distance)
```

```
Minknowski distance between point_1 and point_2 is: 12.0
Manhattan distance between point_1 and point_2 is: 12
```

Note :-when the order is 1, both Minkowski and Manhattan Distance are the same

```python
#Let's calculate the Minkowski Distance formula of order 2

#importing the library
from scipy.spatial import distance

#Defining the points

point_1=(4,5,6)
point_2=(8,9,10)

#computing the minknowski distance

minkowski_distance = distance.minkowski(point_1, point_2, p=2)
print("Minknowski distance between point_1 and point_2 is:",minkowski_distance)

#computing the euclidean distance

Euclidean_distance = distance.euclidean(point_1,point_2)
print("Euclidean distance between point_1 and point_2 is:",Euclidean_distance)
```

```
Minknowski distance between point_1 and point_2 is: 6.928203230275509
Euclidean distance between point_1 and point_2 is: 6.928203230275509
```

Note :-When the order is 2,both Minkowski and Euclidean distances are the same.