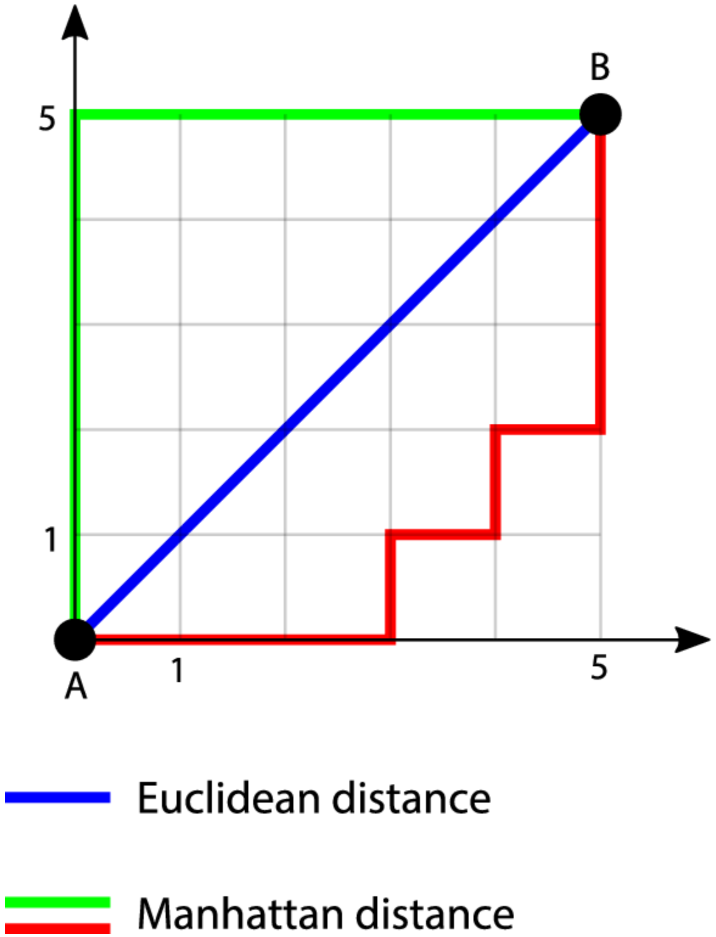
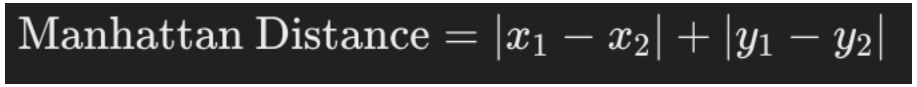
Distance between data points



Manhattan Distance:

Manhattan distance, also known as L1 distance or city block distance, measures the sum of the absolute differences between corresponding coordinates of two data points.

It is calculated as the sum of the absolute differences of the coordinates.



$$\text{Manhattan Distance} = |x_1 - x_2| + |y_1 - y_2|$$

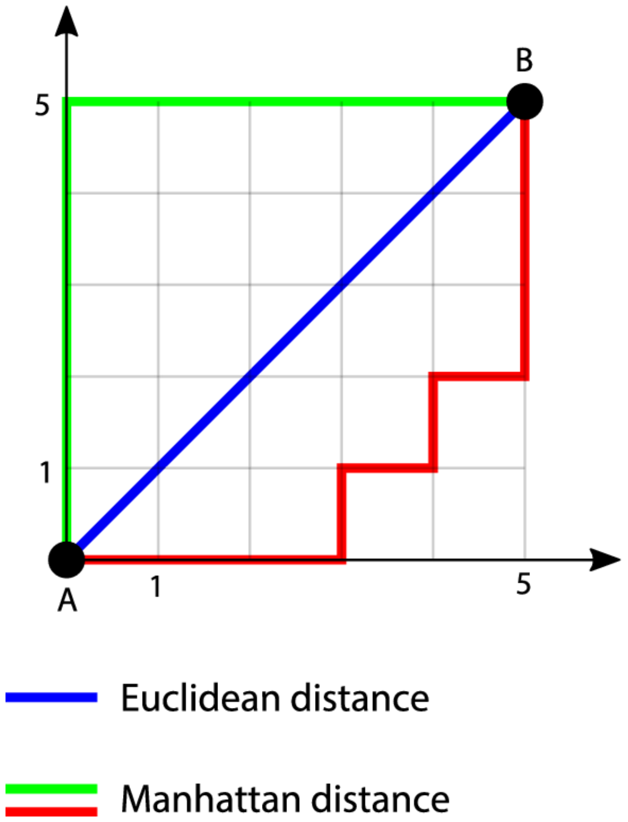
Manhatten python code

```
def manhattan_distance(point1, point2):
        return sum(abs(x - y) for x, y in zip(point1, point2))
```

```
point1 = (1, 2) point2 = (4, 6)
```

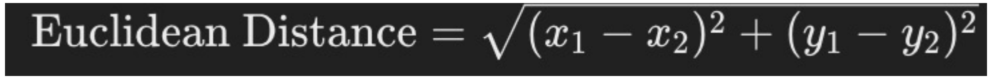
```
print("Manhattan Distance:", manhattan_distance(point1, point2))
```

Euclidean Distance



Euclidean distance, also known as L2 distance, measures the straight-line distance between two points in Euclidean space.

It is calculated as the square root of the sum of the squares of the differences of the coordinates.



$$\text{Euclidean Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

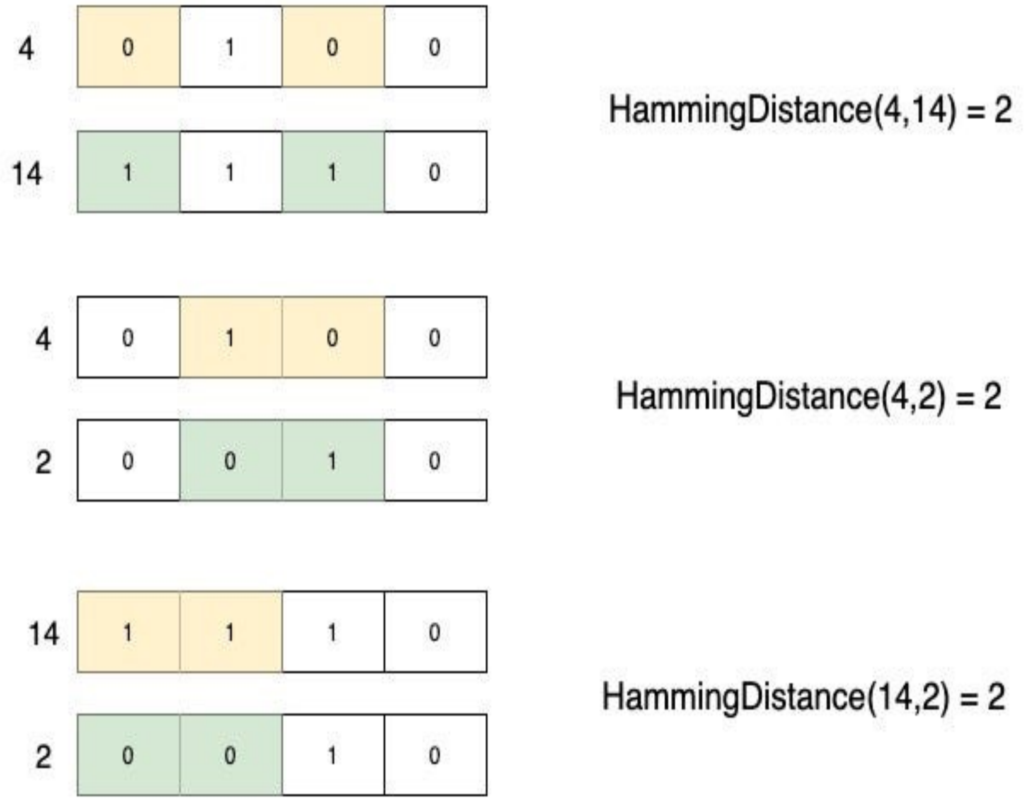
Euclidean python code

```
import math

def euclidean_distance(point1, point2):
        return math.sqrt(sum((x - y) ** 2 for x, y in zip(point1,
point2)))


point1 = (1, 2) point2 = (4, 6)
print("Euclidean Distance:", euclidean_distance(point1, point2))
```

Hamming distance



4	0	1	0	0
14	1	1	1	0

HammingDistance(4,14) = 2

4	0	1	0	0
2	0	0	1	0

HammingDistance(4,2) = 2

14	1	1	1	0
2	0	0	1	0

HammingDistance(14,2) = 2

Hamming distance measures the number of positions at which the corresponding symbols of two strings (or vectors) are different.

It is calculated as the number of positions at which the symbols differ.

Hammer python code

```
def hamming_distance(string1, string2):
        return sum(s1 != s2 for s1, s2 in zip(string1, string2))

string1 = "karolin"
string2 = "kathrin”

print("Hamming Distance:", hamming_distance(string1, string2))
```