

How can the Hamming Distance be calculated in Python? Can you provide some examples?

Authored by
stats writer

April 23, 2024

RECOMMENDED CITATION

stats writer (2024). *How can the Hamming Distance be calculated in Python? Can you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=138480>

The Hamming Distance is a measure of the difference between two strings of equal length. It is calculated by counting the number of positions where the characters in the strings differ. In Python, the Hamming Distance can be calculated by using the built-in function "hamming" from the "scipy.spatial.distance" library. This function takes two strings as input and returns the Hamming Distance between them. For example, if we have two strings "abcd" and "abef", the Hamming Distance would be 2 since there are two positions where the characters differ (c and e). Another example would be the strings "hello" and "hallo", where the Hamming Distance would be 1. Overall, the Hamming Distance calculation in Python is a straightforward process using the "hamming" function and can be useful in various applications such as error detection and correction.

Calculate Hamming Distance in Python (With Examples)

The Hamming distance between two vectors is simply the sum of corresponding elements that differ between the vectors.

For example, suppose we have the following two vectors:

x =

y =

The Hamming distance between the two vectors would be 2, since this is the total number of corresponding elements that have different values.

To calculate the Hamming distance between two arrays

in Python we can use the function from the `scipy.spatial.distance` library, which uses the following syntax:

```
scipy.spatial.distance.hamming(array1, array2)
```

Note that this function returns the percentage of corresponding elements that differ between the two arrays.

Thus, to obtain the Hamming distance we can simply multiply by the length of one of the arrays:

```
scipy.spatial.distance.hamming(array1, array2) *  
len(array1)
```

This tutorial provides several examples of how to use this function in practice.

Example 1: Hamming Distance Between Binary Arrays

The following code shows how to calculate the Hamming distance between two arrays that each contain only two possible values:

```
from scipy.spatial.distance import hamming
```

```
#define arrays
```

```
x =
```

```
y =
```

```
#calculate Hamming distance between the two arrays  
hamming(x, y) * len(x)
```

2.0

The Hamming distance between the two arrays is 2.

Example 2: Hamming Distance Between Numerical Arrays

The following code shows how to calculate the Hamming distance between two arrays that each contain several numerical values:

```
from scipy.spatial.distance import hamming
```

```
#define arrays
```

```
x =
```

```
y =
```

```
#calculate Hamming distance between the two arrays  
hamming(x, y) * len(x)
```

3.0

Example 3: Hamming Distance Between String Arrays

The following code shows how to calculate the Hamming distance between two arrays that each contain several character values:

```
from scipy.spatial.distance import hamming
```

```
#define arrays
```

```
x =
```

```
y =
```

```
#calculate Hamming distance between the two arrays
```

```
hamming(x, y) * len(x)
```

```
1.0
```

The Hamming distance between the two arrays is 1.