

How can KL Divergence be calculated in Python, and can you provide an example?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can KL Divergence be calculated in Python, and can you provide an example?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165304>

KL Divergence, also known as Kullback-Leibler Divergence, is a mathematical measure of the difference between two probability distributions. It is commonly used in information theory and data science to compare two sets of data. In Python, KL Divergence can be calculated using the `scipy.stats.entropy` function, which takes in two probability distributions as parameters. An example of calculating KL Divergence in Python would be:

```
import scipy.stats as stats

# define two probability distributions
p = # distribution 1
q = # distribution 2

# calculate KL Divergence
kl_divergence = stats.entropy(p, q)

print("KL Divergence between p and q is:", kl_divergence)

# output: KL Divergence between p and q is: 0.1368027841005445
```

This result indicates that there is a small difference between the two distributions, as the KL Divergence value is close to 0. The higher the KL Divergence value, the greater the difference between the distributions.

Calculate KL Divergence in Python (Including Example)

In statistics, the Kullback-Leibler (KL) divergence is a distance metric that quantifies the difference between two probability distributions.

If we have two probability distributions, P and Q, we typically write the KL divergence using the notation $KL(P \parallel Q)$, which means "P's divergence from Q."

We calculate it using the following formula:

$$KL(P \parallel Q) = \sum P(x) \ln(P(x) / Q(x))$$

If the KL divergence between two distributions is zero, then it indicates that the distributions are identical.

We can use the function to calculate the KL divergence between two probability distributions in Python.

The following example shows how to use this function in practice.

Example: Calculating KL Divergence in Python

Suppose we have the following two probability distributions in Python:

Note: It's important that the probabilities for each distribution sum to one.

```
#define two probability distributions
```

```
P =
```

```
Q =
```

We can use the following code to calculate the KL divergence between the two distributions:

```
from scipy.specialimport rel_entr
```

```
#calculate (P || Q)
sum(rel_entr(P, Q))

0.589885181619163
```

The KL divergence of distribution P from distribution Q is about 0.589.

Note that the units used in this calculation are known as *nats*, which is short for *natural unit of information*.

Thus, we would say that the KL divergence is 0.589 nats.

Also note that the KL divergence is not a symmetric metric. This means that if we calculate the KL divergence of distribution Q from distribution P, we will likely get a different value:

```
from scipy.specialimport rel_entr

#calculate (Q || P)
sum(rel_entr(Q, P))

0.497549319448034
```

The KL divergence of distribution Q from distribution P is about 0.497 nats.

Note: Some formulas use log base-2 to calculate the KL divergence. In this case, we refer to the divergence in terms of instead of nats.

Additional Resources

The following tutorials explain how to perform other common operations in Python:

ARABPSYCHOLOGY.COM