Cross entropy is a concept used in information theory and machine learning to measure the difference between two probability distributions or the difference between the predicted and actual distributions in classification problems.

In the context of classification, cross entropy is commonly used as a loss function to measure the dissimilarity between the predicted probabilities assigned by a model and the actual probability distribution of the labels.

Here's how cross entropy is defined mathematically:

Given two probability distributions P and Q, where P represents the true distribution (e.g., one-hot encoded ground truth labels) and Q represents the predicted distribution (e.g., softmax output of a neural network), the cross entropy $H(P, Q)$ is calculated as:

$$H(P, Q) = -\sum_i P(i) \log(Q(i))$$

Where:

- $P(i)$ is the probability of the true label i according to the true distribution.
- $Q(i)$ is the probability of the true label i according to the predicted distribution.

Here's an example to illustrate cross entropy in the context of classification:

Let's say we have a classification problem with three classes: cat, dog, and bird. We have a set of ground truth labels:

$\text{Ground Truth Labels: } [1, 0, 0]$ (indicating cat)

And our model predicts the following probabilities for each class:

$\text{Predicted Probabilities: } [0.8, 0.1, 0.1]$ (indicating the model is most confident about the cat class)

To calculate the cross entropy between the ground truth and predicted distributions, we plug these values into the cross entropy formula:

$$H(P, Q) = -\sum_i P(i) \log(Q(i))$$

$$H([1, 0, 0], [0.8, 0.1, 0.1]) = -(1 \times \log(0.8) + 0 \times \log(0.1) + 0 \times \log(0.1))$$

$$= -(0 \times \log(0.8)) = -\log(0.8)$$

In this case, the cross entropy is simply the negative log of the predicted probability of the true class, which measures how well the predicted probability aligns with the ground truth. The lower the cross entropy, the better the model's prediction matches the true distribution.