sklearn.metrics.**confusion\_matrix**(*y\_true*, *y\_pred*, *labels=None*, *sample\_weight=None*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/ef5cb84a/sklearn/metrics/classification.py#L186)[¶](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html#sklearn.metrics.confusion_matrix)

Compute confusion matrix to evaluate the accuracy of a classification

By definition a confusion matrix Cis such that C_{i, j}is equal to the number of observations known to be in group ibut predicted to be in group j.

Thus in binary classification, the count of true negatives is C_{0,0}, false negatives is C_{1,0}, true positives is C_{1,1}and false positives is C_{0,1}.

Read more in the [User Guide](http://scikit-learn.org/stable/modules/model_evaluation.html#confusion-matrix).

|  |  |
| --- | --- |
| **Parameters:** | **y\_true** : array, shape = [n\_samples]  Ground truth (correct) target values.  **y\_pred** : array, shape = [n\_samples]  Estimated targets as returned by a classifier.  **labels** : array, shape = [n\_classes], optional  List of labels to index the matrix. This may be used to reorder or select a subset of labels. If none is given, those that appear at least once in y\_true or y\_pred are used in sorted order.  **sample\_weight** : array-like of shape = [n\_samples], optional  Sample weights. |
| **Returns:** | **C** : array, shape = [n\_classes, n\_classes]  Confusion matrix |

**References**

|  |  |
| --- | --- |
| [[R212]](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html#id1) | [Wikipedia entry for the Confusion matrix](https://en.wikipedia.org/wiki/Confusion_matrix) |

**Examples**

>>>

**>>> from** **sklearn.metrics** **import** confusion\_matrix

**>>>** y\_true = [2, 0, 2, 2, 0, 1]

**>>>** y\_pred = [0, 0, 2, 2, 0, 2]

**>>>** confusion\_matrix(y\_true, y\_pred)

array([[2, 0, 0],

[0, 0, 1],

[1, 0, 2]])

>>>

**>>>** y\_true = ["cat", "ant", "cat", "cat", "ant", "bird"]

**>>>** y\_pred = ["ant", "ant", "cat", "cat", "ant", "cat"]

**>>>** confusion\_matrix(y\_true, y\_pred, labels=["ant", "bird", "cat"])

array([[2, 0, 0],

[0, 0, 1],

[1, 0, 2]])

In the binary case, we can extract true positives, etc as follows:

>>>

**>>>** tn, fp, fn, tp = confusion\_matrix([0, 1, 0, 1], [1, 1, 1, 0]).ravel()

**>>>** (tn, fp, fn, tp)

(0, 2, 1, 1)