f1_score

sklearn.metrics.f1_score(*y_true*, *y_pred*, *, *labels=None*, *pos_label=1*, *average='binary'*, *sample_weight=None*, *zero_division='warn'*) [source]

Compute the F1 score, also known as balanced F-score or F-measure.

The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

$$\text{F1} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}$$

Where TP is the number of true positives, FN is the number of false negatives, and FP is the number of false positives. F1 is by default calculated as 0.0 when there are no true positives, false negatives, or false positives.

Support beyond binary targets is achieved by treating multiclass and multilabel data as a collection of binary problems, one for each label. For the binary case, setting `average='binary'` will return F1 score for `pos_label`. If `average` is not `'binary'`, `pos_label` is ignored and F1 score for both classes are computed, then averaged or both returned (when `average=None`). Similarly, for multiclass and multilabel targets, F1 score for all `labels` are either returned or averaged depending on the `average` parameter. Use `labels` specify the set of labels to calculate F1 score for.

Read more in the User Guide.

Parameters:

y_true : *1d array-like, or label indicator array / sparse matrix*

Ground truth (correct) target values.

y_pred : *1d array-like, or label indicator array / sparse matrix*

Estimated targets as returned by a classifier.

labels : *array-like, default=None*

The set of labels to include when `average != 'binary'`, and their order if `average is None`. Labels present in the data can be excluded, for example in multiclass classification to exclude a “negative class”. Labels not present in the data can be included and will be

"assigned" 0 samples. For multilabel targets, labels are column indices. By default, all labels in `y_true` and `y_pred` are used in sorted order.

> ***Changed in version 0.17:*** Parameter `labels` improved for multiclass problem.

pos_label : *int, float, bool or str, default=1*

The class to report if `average='binary'` and the data is binary, otherwise this parameter is ignored. For multiclass or multilabel targets, set `labels=[pos_label]` and `average != 'binary'` to report metrics for one label only.

average : *{'micro', 'macro', 'samples', 'weighted', 'binary'} or None, default='binary'*

This parameter is required for multiclass/multilabel targets. If `None`, the metrics for each class are returned. Otherwise, this determines the type of averaging performed on the data:

`'binary'`:

Only report results for the class specified by `pos_label`. This is applicable only if targets (`y_{true,pred}`) are binary.

`'micro'`:

Calculate metrics globally by counting the total true positives, false negatives and false positives.

`'macro'`:

Calculate metrics for each label, and find their unweighted mean. This does not take label imbalance into account.

`'weighted'`:

Calculate metrics for each label, and find their average weighted by support (the number of true instances for each label). This alters 'macro' to account for label imbalance; it can result in an F-score that is not between precision and recall.

`'samples'`:

Calculate metrics for each instance, and find their average (only meaningful for multilabel classification where this differs from `accuracy_score`).

sample_weight : *array-like of shape (n_samples,), default=None*

Sample weights.

zero_division : *{"warn", 0.0, 1.0, np.nan}, default="warn"*

Sets the value to return when there is a zero division, i.e. when all predictions and labels are negative.

Notes: - If set to “warn”, this acts like 0, but a warning is also raised. - If set to `np.nan`, such values will be excluded from the average.

> **Added in version 1.3:** `np.nan` option was added.

Returns:

f1_score : *float or array of float, shape = [n_unique_labels]*

F1 score of the positive class in binary classification or weighted average of the F1 scores of each class for the multiclass task.

See also

`fbeta_score`
Compute the F-beta score.

`precision_recall_fscore_support`
Compute the precision, recall, F-score, and support.

`jaccard_score`
Compute the Jaccard similarity coefficient score.

`multilabel_confusion_matrix`
Compute a confusion matrix for each class or sample.

Notes

When `true positive + false positive + false negative == 0` (i.e. a class is completely absent from both `y_true` or `y_pred`), f-score is undefined. In such cases, by default f-score will be set to 0.0, and `UndefinedMetricWarning` will be raised. This behavior can be modified by setting the `zero_division` parameter.

References

[1] Wikipedia entry for the F1-score.

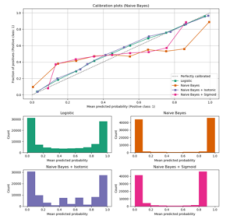
Examples

```
>>> import numpy as np
>>> from sklearn.metrics import f1_score
>>> y_true = [0, 1, 2, 0, 1, 2]
>>> y_pred = [0, 2, 1, 0, 0, 1]
>>> f1_score(y_true, y_pred, average='macro')
0.26...
>>> f1_score(y_true, y_pred, average='micro')
0.33...
>>> f1_score(y_true, y_pred, average='weighted')
0.26...
>>> f1_score(y_true, y_pred, average=None)
array([0.8, 0. , 0. ])
```

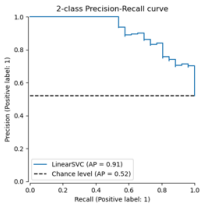
```
>>> # binary classification
>>> y_true_empty = [0, 0, 0, 0, 0, 0]
>>> y_pred_empty = [0, 0, 0, 0, 0, 0]
>>> f1_score(y_true_empty, y_pred_empty)
0.0...
>>> f1_score(y_true_empty, y_pred_empty, zero_division=1.0)
1.0...
>>> f1_score(y_true_empty, y_pred_empty, zero_division=np.nan)
nan...
```

```
>>> # multilabel classification
>>> y_true = [[0, 0, 0], [1, 1, 1], [0, 1, 1]]
>>> y_pred = [[0, 0, 0], [1, 1, 1], [1, 1, 0]]
>>> f1_score(y_true, y_pred, average=None)
array([0.66666667, 1.        , 0.66666667])
```

Gallery examples



Probability Calibration curves



Precision-Recall

Semi-supervised Classification on a Text Dataset