sklearn.metrics.ConfusionMatrixDisplay

class `sklearn.metrics.ConfusionMatrixDisplay`(*confusion_matrix*, ***, *display_labels=None*) [source]

Confusion Matrix visualization.

It is recommend to use `from_estimator` or `from_predictions` to create a `ConfusionMatrixDisplay`. All parameters are stored as attributes.

Read more in the User Guide.

Parameters:

confusion_matrix : *ndarray of shape (n_classes, n_classes)*

Confusion matrix.

display_labels : *ndarray of shape (n_classes,), default=None*

Display labels for plot. If None, display labels are set from 0 to `n_classes - 1`.

Attributes:

im_ : *matplotlib AxesImage*

Image representing the confusion matrix.

text_ : *ndarray of shape (n_classes, n_classes), dtype=matplotlib Text, or None*

Array of matplotlib axes. `None` if `include_values` is false.

ax_ : *matplotlib Axes*

Axes with confusion matrix.

figure_ : *matplotlib Figure*

Figure containing the confusion matrix.

See also:

`confusion_matrix`

Compute Confusion Matrix to evaluate the accuracy of a classification.

`ConfusionMatrixDisplay.from_estimator`

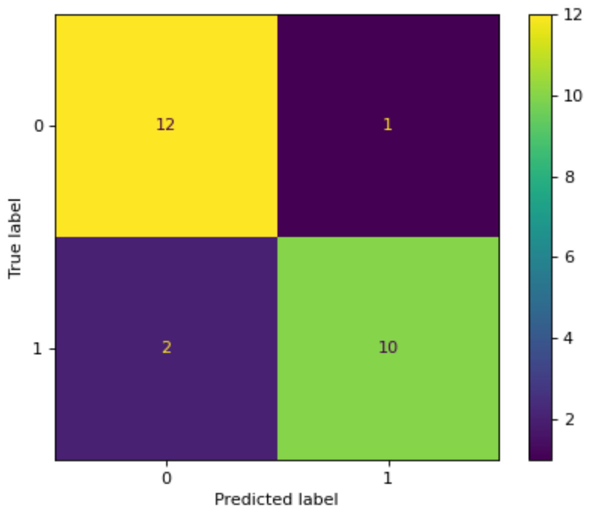
Plot the confusion matrix given an estimator, the data, and the label.

`ConfusionMatrixDisplay.from_predictions`

Plot the confusion matrix given the true and predicted labels.

Examples

```
>>> import matplotlib.pyplot as plt
>>> from sklearn.datasets import make_classification
>>> from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.svm import SVC
>>> X, y = make_classification(random_state=0)
>>> X_train, X_test, y_train, y_test = train_test_split(X, y,
...                                                     random_state=0)
>>> clf = SVC(random_state=0)
>>> clf.fit(X_train, y_train)
SVC(random_state=0)
>>> predictions = clf.predict(X_test)
>>> cm = confusion_matrix(y_test, predictions, labels=clf.classes_)
>>> disp = ConfusionMatrixDisplay(confusion_matrix=cm,
...                               display_labels=clf.classes_)
>>> disp.plot()
<...>
>>> plt.show()
```



Methods

from_estimator(estimator, X, y, *[, labels, ...])	Plot Confusion Matrix given an estimator and some data.
from_predictions(y_true, y_pred, *[, ...])	Plot Confusion Matrix given true and predicted labels.
plot(*[, include_values, cmap, ...])	Plot visualization.

classmethod `from_estimator`(*estimator*, *X*, *y*, *, *labels=None*, *sample_weight=None*, *normalize=None*, *display_labels=None*, *include_values=True*, *xticks_rotation='horizontal'*, *values_format=None*, *cmap='viridis'*, *ax=None*, *colorbar=True*, *im_kw=None*) [source]

Plot Confusion Matrix given an estimator and some data.

Read more in the User Guide.

New in version 1.0.

Parameters:

estimator : ***estimator instance***

Fitted classifier or a fitted `Pipeline` in which the last estimator is a classifier.

X : ***{array-like, sparse matrix} of shape (n_samples, n_features)***

Input values.

y : ***array-like of shape (n_samples,)***

Target values.

labels : ***array-like of shape (n_classes,), default=None***

List of labels to index the confusion 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,), default=None***

Sample weights.

normalize : ***{'true', 'pred', 'all'}, default=None***

Either to normalize the counts display in the matrix:

- if `'true'`, the confusion matrix is normalized over the true conditions (e.g. rows);
- if `'pred'`, the confusion matrix is normalized over the predicted conditions (e.g. columns);
- if `'all'`, the confusion matrix is normalized by the total number of samples;
- if `None` (default), the confusion matrix will not be normalized.

display_labels : ***array-like of shape (n_classes,), default=None***

Target names used for plotting. By default, `labels` will be used if it is defined, otherwise the unique labels of `y_true` and `y_pred` will be used.

include_values : ***bool, default=True***

Includes values in confusion matrix.

xticks_rotation : ***{'vertical', 'horizontal'} or float, default='horizontal'***

Rotation of xtick labels.

values_format : ***str, default=None***

Format specification for values in confusion matrix. If `None`, the format specification is 'd' or '.2g' whichever is shorter.

cmap : ***str or matplotlib Colormap, default='viridis'***

Colormap recognized by matplotlib.

ax : ***matplotlib Axes, default=None***

Axes object to plot on. If `None`, a new figure and axes is created.

colorbar : ***bool, default=True***

Whether or not to add a colorbar to the plot.

im_kw : ***dict, default=None***

Dict with keywords passed to `matplotlib.pyplot.imshow` call.

Returns:

display : *`ConfusionMatrixDisplay`*

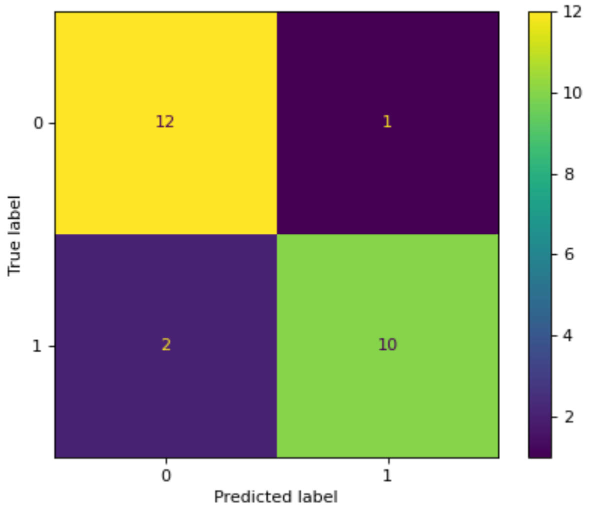
See also:

`ConfusionMatrixDisplay.from_predictions`

Plot the confusion matrix given the true and predicted labels.

Examples

```
>>> import matplotlib.pyplot as plt
>>> from sklearn.datasets import make_classification
>>> from sklearn.metrics import ConfusionMatrixDisplay
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.svm import SVC
>>> X, y = make_classification(random_state=0)
>>> X_train, X_test, y_train, y_test = train_test_split(
...         X, y, random_state=0)
>>> clf = SVC(random_state=0)
>>> clf.fit(X_train, y_train)
SVC(random_state=0)
>>> ConfusionMatrixDisplay.from_estimator(
...     clf, X_test, y_test)
<...>
>>> plt.show()
```



classmethod **`from_predictions`**(*y_true*, *y_pred*, ***, *labels=None*, *sample_weight=None*, *normalize=None*, *display_labels=None*, *include_values=True*, *xticks_rotation='horizontal'*, *values_format=None*, *cmap='viridis'*, *ax=None*, *colorbar=True*, *im_kw=None*) [source]

Plot Confusion Matrix given true and predicted labels.

Read more in the User Guide.

New in version 1.0.

Parameters:

y_true : *array-like of shape (n_samples,)*
True labels.

y_pred : *array-like of shape (n_samples,)*
The predicted labels given by the method `predict` of an classifier.

labels : *array-like of shape (n_classes,), default=None*
List of labels to index the confusion 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,), default=None*
Sample weights.

normalize : *{'true', 'pred', 'all'}, default=None*
Either to normalize the counts display in the matrix:

- if `'true'`, the confusion matrix is normalized over the true conditions (e.g. rows);
- if `'pred'`, the confusion matrix is normalized over the predicted conditions (e.g. columns);
- if `'all'`, the confusion matrix is normalized by the total number of samples;
- if `None` (default), the confusion matrix will not be normalized.

display_labels : *array-like of shape (n_classes,), default=None*
Target names used for plotting. By default, `labels` will be used if it is defined, otherwise the unique labels of `y_true` and `y_pred` will be used.

include_values : *bool, default=True*
Includes values in confusion matrix.

xticks_rotation : *{'vertical', 'horizontal'} or float, default='horizontal'*
Rotation of xtick labels.

values_format : *str, default=None*
Format specification for values in confusion matrix. If `None`, the format specification is 'd' or '.2g' whichever is shorter.

cmap : *str or matplotlib Colormap, default='viridis'*
Colormap recognized by matplotlib.

ax : *matplotlib Axes, default=None*
Axes object to plot on. If `None`, a new figure and axes is created.

colorbar : *bool, default=True*
Whether or not to add a colorbar to the plot.

im_kw : *dict, default=None*
Dict with keywords passed to `matplotlib.pyplot.imshow` call.

Returns:

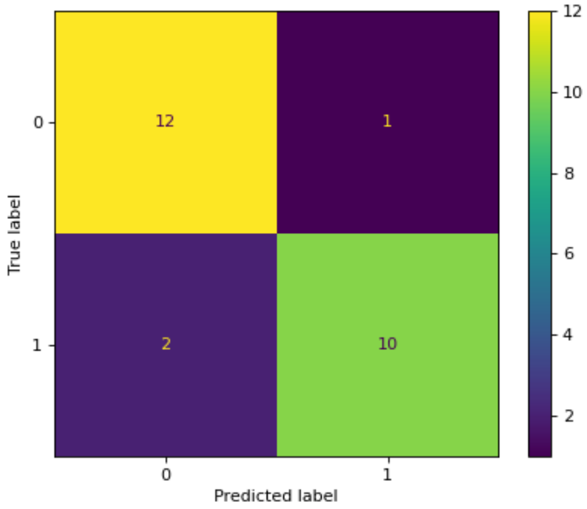
display : *`ConfusionMatrixDisplay`*

See also:

`ConfusionMatrixDisplay.from_estimator`
Plot the confusion matrix given an estimator, the data, and the label.

Examples

```
>>> import matplotlib.pyplot as plt
>>> from sklearn.datasets import make_classification
>>> from sklearn.metrics import ConfusionMatrixDisplay
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.svm import SVC
>>> X, y = make_classification(random_state=0)
>>> X_train, X_test, y_train, y_test = train_test_split(
...         X, y, random_state=0)
>>> clf = SVC(random_state=0)
>>> clf.fit(X_train, y_train)
SVC(random_state=0)
>>> y_pred = clf.predict(X_test)
>>> ConfusionMatrixDisplay.from_predictions(
...    y_test, y_pred)
<...>
>>> plt.show()
```



plot(**, include_values=True, cmap='viridis', xticks_rotation='horizontal', values_format=None, ax=None, colorbar=True, im_kw=None*) [source]

Plot visualization.

Parameters:

include_values : *bool, default=True*

Includes values in confusion matrix.

cmap : *str or matplotlib Colormap, default='viridis'*

Colormap recognized by matplotlib.

xticks_rotation : *{'vertical', 'horizontal'} or float, default='horizontal'*

Rotation of xtick labels.

values_format : *str, default=None*

Format specification for values in confusion matrix. If `None`, the format specification is 'd' or '.2g' whichever is shorter.

ax : *matplotlib axes, default=None*

Axes object to plot on. If `None`, a new figure and axes is created.

colorbar : *bool, default=True*

Whether or not to add a colorbar to the plot.

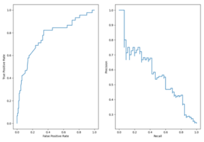
im_kw : *dict, default=None*

Dict with keywords passed to `matplotlib.pyplot.imshow` call.

Returns:

display : *`ConfusionMatrixDisplay`*

Examples using `sklearn.metrics.ConfusionMatrixDisplay`



Visualizations with Display Objects

Show this page source

Toggle Menu