

`sklearn.preprocessing`.LabelEncoder

`class sklearn.preprocessing.LabelEncoder`

[\[source\]](#)

Encode target labels with value between 0 and `n_classes-1`.

This transformer should be used to encode target values, *i.e.* `y`, and not the input `x`.

Read more in the [User Guide](#).

New in version 0.12.

Attributes:

`classes_ : ndarray of shape (n_classes,)`

Holds the label for each class.

See also:

[OrdinalEncoder](#)

Encode categorical features using an ordinal encoding scheme.

[OneHotEncoder](#)

Encode categorical features as a one-hot numeric array.

Examples

LabelEncoder can be used to normalize labels.

```
>>> from sklearn import preprocessing
>>> le = preprocessing.LabelEncoder()
>>> le.fit([1, 2, 2, 6])
LabelEncoder()
>>> le.classes_
array([1, 2, 6])
>>> le.transform([1, 1, 2, 6])
array([0, 0, 1, 2]...)
>>> le.inverse_transform([0, 0, 1, 2])
array([1, 1, 2, 6])
```

It can also be used to transform non-numerical labels (as long as they are hashable and comparable) to numerical labels.

```
>>> le = preprocessing.LabelEncoder()
>>> le.fit(["paris", "paris", "tokyo", "amsterdam"])
LabelEncoder()
>>> list(le.classes_)
['amsterdam', 'paris', 'tokyo']
>>> le.transform(["tokyo", "tokyo", "paris"])
array([2, 2, 1]...)
>>> list(le.inverse_transform([2, 2, 1]))
['tokyo', 'tokyo', 'paris']
```

Methods

fit(y)	Fit label encoder.
fit_transform(y)	Fit label encoder and return encoded labels.
get_metadata_routing()	Get metadata routing of this object.
get_params([deep])	Get parameters for this estimator.
inverse_transform(y)	Transform labels back to original encoding.
set_output(*[, transform])	Set output container.
set_params(**params)	Set the parameters of this estimator.
transform(y)	Transform labels to normalized encoding.

Toggle Menu

fit(y)
[\[source\]](#)

Fit label encoder.

Parameters:

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

Returns:

self : *returns an instance of self.*
Fitted label encoder.

fit_transform(y)
[\[source\]](#)

Fit label encoder and return encoded labels.

Parameters:

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

Returns:

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

get_metadata_routing()
[\[source\]](#)

Get metadata routing of this object.

Please check [User Guide](#) on how the routing mechanism works.

Returns:

routing : *MetadataRequest*
A MetadataRequest encapsulating routing information.

get_params(deep=True)
[\[source\]](#)

Get parameters for this estimator.

Parameters:

deep : *bool, default=True*
If True, will return the parameters for this estimator and contained subobjects that are estimators.

Returns:

params : *dict*
Parameter names mapped to their values.

inverse_transform(y)
[\[source\]](#)

Transform labels back to original encoding.

Parameters:

y : *ndarray of shape (n_samples,)*
Target values.

Returns:

y : *ndarray of shape (n_samples,)*
Original encoding.

[Toggle Menu](#)

```
set_output(*, transform=None)
```

[\[source\]](#)

Set output container.

See [Introducing the set_output API](#) for an example on how to use the API.

Parameters:

transform : {"default", "pandas"}, default=None

Configure output of transform and fit_transform.

- "default": Default output format of a transformer
- "pandas": DataFrame output
- None: Transform configuration is unchanged

Returns:

self : *estimator instance*

Estimator instance.

```
set_params(**params)
```

[\[source\]](#)

Set the parameters of this estimator.

The method works on simple estimators as well as on nested objects (such as [Pipeline](#)). The latter have parameters of the form <component>__<parameter> so that it's possible to update each component of a nested object.

Parameters:

****params** : dict

Estimator parameters.

Returns:

self : *estimator instance*

Estimator instance.

```
transform(y)
```

[\[source\]](#)

Transform labels to normalized encoding.

Parameters:

y : array-like of shape (n_samples,)

Target values.

Returns:

y : array-like of shape (n_samples,)

Labels as normalized encodings.

© 2007 - 2023, scikit-learn developers (BSD License). [Show this page source](#)