
sklearn-features Documentation

Release 0.0.2

Tim Leslie

Nov 15, 2017

Contents:

1	API summary	3
1.1	transformers.DataFrameSelector	3
1.2	transformers.SeriesReshaper	4
1.3	transformers.DataFrameReshaper	4
1.4	transformers.NullTransformer	5
1.5	transformers.EqualitySelector	5
1.6	transformers.ScalingTransformer	6
1.7	transformers.DateAttributeTransformer	6
1.8	transformers.DateMethodTransformer	7
1.9	transformers.MultiDateTransformer	7
1.10	transformers.LinearDateTransformer	8
1.11	transformers.LabelEncoderWithUnknown	8
1.12	transformers.OneHotWithUnknown	9
1.13	transformers.OneHotWithFixedFeatures	9
1.14	transformers.OneHotWithFixedFeatureDict	10
1.15	transformers.series_pipeline	10
1.16	transformers.dataframe_pipeline	10
2	Indices and tables	11

sklearn-features provides an API to simplify feature engineering with scikit-learn and pandas.

CHAPTER 1

API summary

<code>transformers.DataFrameSelector(key)</code>	Transforms a DataFrame into a Series by selecting a single column by key.
<code>transformers.SeriesReshaper</code>	Transforms a Series of size N into an (N, 1) shaped numpy array.
<code>transformers.DataFrameReshaper</code>	Transforms a DataFrame of size NxM into an (N, M) shaped numpy array.
<code>transformers.NullTransformer([feature_name])</code>	Pass through the Series completely unchanged.
<code>transformers.EqualitySelector(target)</code>	
<code>transformers.ScalingTransformer(scaling_factor)</code>	Apply a constant scaling factor to a Series.
<code>transformers.DateAttributeTransformer(attr)</code>	Select a particular attribute from the .dt property of a Series.
<code>transformers.DateMethodTransformer(method, ...)</code>	Execute a particular method from the .dt property of a Series.
<code>transformers.MultiDateTransformer(dates)</code>	
<code>transformers.LinearDateTransformer([d0, delta])</code>	Convert a datetime Series into a float Series.
<code>transformers.LabelEncoderWithUnknown([...])</code>	Convert a categorical feature into values [0, n], where [0, n] represent the known categories from the training data and n represents unknown data.
<code>transformers.OneHotWithUnknown([feature_names])</code>	
<code>transformers.OneHotWithFixedFeatures([...])</code>	
<code>transformers.OneHotWithFixedFeatureDict(...)</code>	
<code>transformers.series_pipeline(key, steps)</code>	
<code>transformers.dataframe_pipeline(key, steps)</code>	

1.1 transformers.DataFrameSelector

class `transformers.DataFrameSelector` (*key*)
Transforms a DataFrame into a Series by selecting a single column by key.

`__init__` (*key*)

Methods

<code>__init__(key)</code>	
<code>fit(df[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(df)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.2 transformers.SeriesReshaper

class `transformers.SeriesReshaper`

Transforms a Series of size N into an (N, 1) shaped numpy array.

`__init__()`
x.`__init__()` initializes x; see `help(type(x))` for signature

Methods

<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.3 transformers.DataFrameReshaper

class `transformers.DataFrameReshaper`

Transforms a DataFrame of size NxM into an (N, M) shaped numpy array.

`__init__()`
x.`__init__()` initializes x; see `help(type(x))` for signature

Methods

<code>fit(df[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(df)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.4 transformers.NullTransformer

class `transformers.NullTransformer` (*feature_name=None*)

Pass through the Series completely unchanged.

`__init__` (*feature_name=None*)

Methods

<code>__init__([feature_name])</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.5 transformers.EqualitySelector

class `transformers.EqualitySelector` (*target*)

`__init__` (*target*)

Methods

<code>__init__(target)</code>	
<code>fit(ds[, y])</code>	

Continued on next page

Table 1.10 – continued from previous page

<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.6 transformers.ScalingTransformer

class `transformers.ScalingTransformer` (*scaling_factor*)
Apply a constant scaling factor to a Series.
`__init__` (*scaling_factor*)

Methods

<code>__init__</code> (<i>scaling_factor</i>)	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.7 transformers.DateAttributeTransformer

class `transformers.DateAttributeTransformer` (*attr*)
Select a particular attribute from the .dt property of a Series.
<https://pandas.pydata.org/pandas-docs/stable/api.html#datetimelike-properties>
`__init__` (*attr*)

Methods

<code>__init__</code> (<i>attr</i>)
Continued on next page

Table 1.14 – continued from previous page

<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.8 transformers.DateMethodTransformer

class `transformers.DateMethodTransformer` (*method, kwargs*)

Execute a particular method from the `.dt` property of a Series.

<https://pandas.pydata.org/pandas-docs/stable/api.html#datetimelike-properties>

`__init__` (*method, kwargs*)

Methods

<code>__init__(method, kwargs)</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.9 transformers.MultiDateTransformer

class `transformers.MultiDateTransformer` (*dates*)

`__init__` (*dates*)

Methods

<code>__init__(dates)</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.10 transformers.LinearDateTransformer

class `transformers.LinearDateTransformer` (*d0=None, delta=Timedelta('1 days 00:00:00')*)

Convert a datetime Series into a float Series.

Perform a linear transformation based on *d0* and *delta*.

Defaults: *d0*: `training_ds.min()` *delta*: 1 day

`__init__` (*d0=None, delta=Timedelta('1 days 00:00:00')*)

Methods

<code>__init__([d0, delta])</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.11 transformers.LabelEncoderWithUnknown

class `transformers.LabelEncoderWithUnknown` (*feature_name=None*)

Convert a categorical feature into values [0, n], where [0, n) represent the known categories from the training data and n represents unknown data.

`__init__` (*feature_name=None*)

Methods

<code>__init__([feature_name])</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.12 transformers.OneHotWithUnknown

class `transformers.OneHotWithUnknown` (*feature_names=None*)

`__init__` (*feature_names=None*)

Methods

<code>__init__([feature_names])</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.13 transformers.OneHotWithFixedFeatures

class `transformers.OneHotWithFixedFeatures` (*feature_names=None*)

`__init__` (*feature_names=None*)

Methods

<code>__init__([feature_names])</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.14 transformers.OneHotWithFixedFeatureDict

`class transformers.OneHotWithFixedFeatureDict` (*feature_name_dict*)

`__init__` (*feature_name_dict*)

Methods

<code>__init__(feature_name_dict)</code>	
<code>fit(ds[, y])</code>	
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_feature_names()</code>	
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>set_params(**params)</code>	Set the parameters of this estimator.
<code>transform(ds)</code>	

Attributes

<code>DEFAULT_PIPELINE_NAME</code>

1.15 transformers.series_pipeline

`transformers.series_pipeline` (*key, steps*)

1.16 transformers.dataframe_pipeline

`transformers.dataframe_pipeline` (*key, steps*)

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

Symbols

`__init__()` (transformers.DataFrameReshaper method), 4
`__init__()` (transformers.DataFrameSelector method), 3
`__init__()` (transformers.DateAttributeTransformer method), 6
`__init__()` (transformers.DateMethodTransformer method), 7
`__init__()` (transformers.EqualitySelector method), 5
`__init__()` (transformers.LabelEncoderWithUnknown method), 8
`__init__()` (transformers.LinearDateTransformer method), 8
`__init__()` (transformers.MultiDateTransformer method), 7
`__init__()` (transformers.NullTransformer method), 5
`__init__()` (transformers.OneHotWithFixedFeatureDict method), 10
`__init__()` (transformers.OneHotWithFixedFeatures method), 9
`__init__()` (transformers.OneHotWithUnknown method), 9
`__init__()` (transformers.ScalingTransformer method), 6
`__init__()` (transformers.SeriesReshaper method), 4

D

`dataframe_pipeline()` (in module transformers), 10
DataFrameReshaper (class in transformers), 4
DataFrameSelector (class in transformers), 3
DateAttributeTransformer (class in transformers), 6
DateMethodTransformer (class in transformers), 7

E

EqualitySelector (class in transformers), 5

L

LabelEncoderWithUnknown (class in transformers), 8
LinearDateTransformer (class in transformers), 8

M

MultiDateTransformer (class in transformers), 7

N

NullTransformer (class in transformers), 5

O

OneHotWithFixedFeatureDict (class in transformers), 10
OneHotWithFixedFeatures (class in transformers), 9
OneHotWithUnknown (class in transformers), 9

S

ScalingTransformer (class in transformers), 6
`series_pipeline()` (in module transformers), 10
SeriesReshaper (class in transformers), 4