sklearn-features Documentation

Release 0.0.2

Tim Leslie

Nov 15, 2017

Contents:

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

CHAPTER 1

API summary

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

1.1 transformers.DataFrameSelector

class transformers.**DataFrameSelector**(*key*)

Transforms a DataFrame into a Series by selecting a single column by key.

__init__(*key*)

Methods

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

Attributes

`DEFAULT_PIPELINE_NAME`

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

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

Attributes

`DEFAULT_PIPELINE_NAME`

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

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

Attributes

DEFAULT_PIPELINE_NAME

1.4 transformers.NullTransformer

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

Pass through the Series completely unchanged.

__init__(*feature_name=None*)

Methods

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

Attributes

DEFAULT_PIPELINE_NAME

1.5 transformers.EqualitySelector

class transformers.**EqualitySelector**(*target*)

__init__(*target*)

Methods

__init__(target)
fit(ds[, y])

Continued on next page

Table 1.10 – continued from previous page

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

Attributes

DEFAULT_PIPELINE_NAME

1.6 transformers.ScalingTransformer

class transformers.**ScalingTransformer**(*scaling_factor*)

Apply a constant scaling factor to a Series.

__init__(*scaling_factor*)

Methods

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

Attributes

DEFAULT_PIPELINE_NAME

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

__init__(attr)

Continued on next page

Table 1.14 – continued from previous page

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

Attributes

DEFAULT_PIPELINE_NAME

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

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

Attributes

DEFAULT_PIPELINE_NAME

1.9 transformers.MultiDateTransformer

class transformers.**MultiDateTransformer**(*dates*)

__init__(*dates*)

Methods

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

Attributes

DEFAULT_PIPELINE_NAME

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

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

Attributes

DEFAULT_PIPELINE_NAME

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

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

Attributes

DEFAULT_PIPELINE_NAME

1.12 transformers.OneHotWithUnknown

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

__init__(*feature_names=None*)

Methods

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

Attributes

DEFAULT_PIPELINE_NAME

1.13 transformers.OneHotWithFixedFeatures

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

__init__(*feature_names=None*)

Methods

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

Attributes

DEFAULT_PIPELINE_NAME

1.14 transformers.OneHotWithFixedFeatureDict

class transformers.**OneHotWithFixedFeatureDict**(*feature_name_dict*)

__init__(*feature_name_dict*)

Methods

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

Attributes

DEFAULT_PIPELINE_NAME

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

Index

Symbols

D

E

L

M

N

O

S