Regression Exercise

May 22, 2022

[1]:

```
# load data
import pandas as pd
df = pd.read_csv("HousingData.csv")
```

[2]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     486 non-null    float64
 1   ZN       486 non-null    float64
 2   INDUS    486 non-null    float64
 3   CHAS     486 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      486 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64
 9   TAX      506 non-null    int64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    486 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(12), int64(2)
memory usage: 55.5 KB
```

Where MEDV is the target variable

[3]:

```
df.describe()
```

[3]:

```
             CRIM          ZN       INDUS        CHAS         NOX          RM  \
count  486.000000  486.000000  486.000000  486.000000  506.000000  506.000000
mean     3.611874   11.211934   11.083992    0.069959    0.554695    6.284634
std      8.720192   23.388876    6.835896    0.255340    0.115878    0.702617
min      0.006320    0.000000    0.460000    0.000000    0.385000    3.561000
25%      0.081900    0.000000    5.190000    0.000000    0.449000    5.885500
```

```
50%      0.253715    0.000000    9.690000    0.000000    0.538000    6.208500
75%      3.560263   12.500000   18.100000    0.000000    0.624000    6.623500
max     88.976200  100.000000   27.740000    1.000000    0.871000    8.780000

              AGE         DIS         RAD         TAX     PTRATIO           B  \
count  486.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean    68.518519    3.795043    9.549407  408.237154   18.455534  356.674032
std     27.999513    2.105710    8.707259  168.537116    2.164946   91.294864
min      2.900000    1.129600    1.000000  187.000000   12.600000    0.320000
25%     45.175000    2.100175    4.000000  279.000000   17.400000  375.377500
50%     76.800000    3.207450    5.000000  330.000000   19.050000  391.440000
75%     93.975000    5.188425   24.000000  666.000000   20.200000  396.225000
max    100.000000   12.126500   24.000000  711.000000   22.000000  396.900000

            LSTAT        MEDV
count  486.000000  506.000000
mean    12.715432   22.532806
std      7.155871    9.197104
min      1.730000    5.000000
25%      7.125000   17.025000
50%     11.430000   21.200000
75%     16.955000   25.000000
max     37.970000   50.000000
```

[4]:

```
# lets fill missing values with median
Age_median = df["AGE"].median()
LSTAT_median = df["LSTAT"].median()
CRIM_median = df["CRIM"].median()
ZN_median = df["ZN"].median()
INDUS_median = df["INDUS"].median()
CHAS_median = df["CHAS"].median()
```

[5]:

```
df["AGE"] = df["AGE"].fillna(Age_median)
df["LSTAT"] = df["LSTAT"].fillna(LSTAT_median)
df["CRIM"] = df["AGE"].fillna(CRIM_median)
df["ZN"] = df["ZN"].fillna(ZN_median)
df["INDUS"] = df["INDUS"].fillna(INDUS_median)
df["CHAS"] = df["CHAS"].fillna(CHAS_median)
```

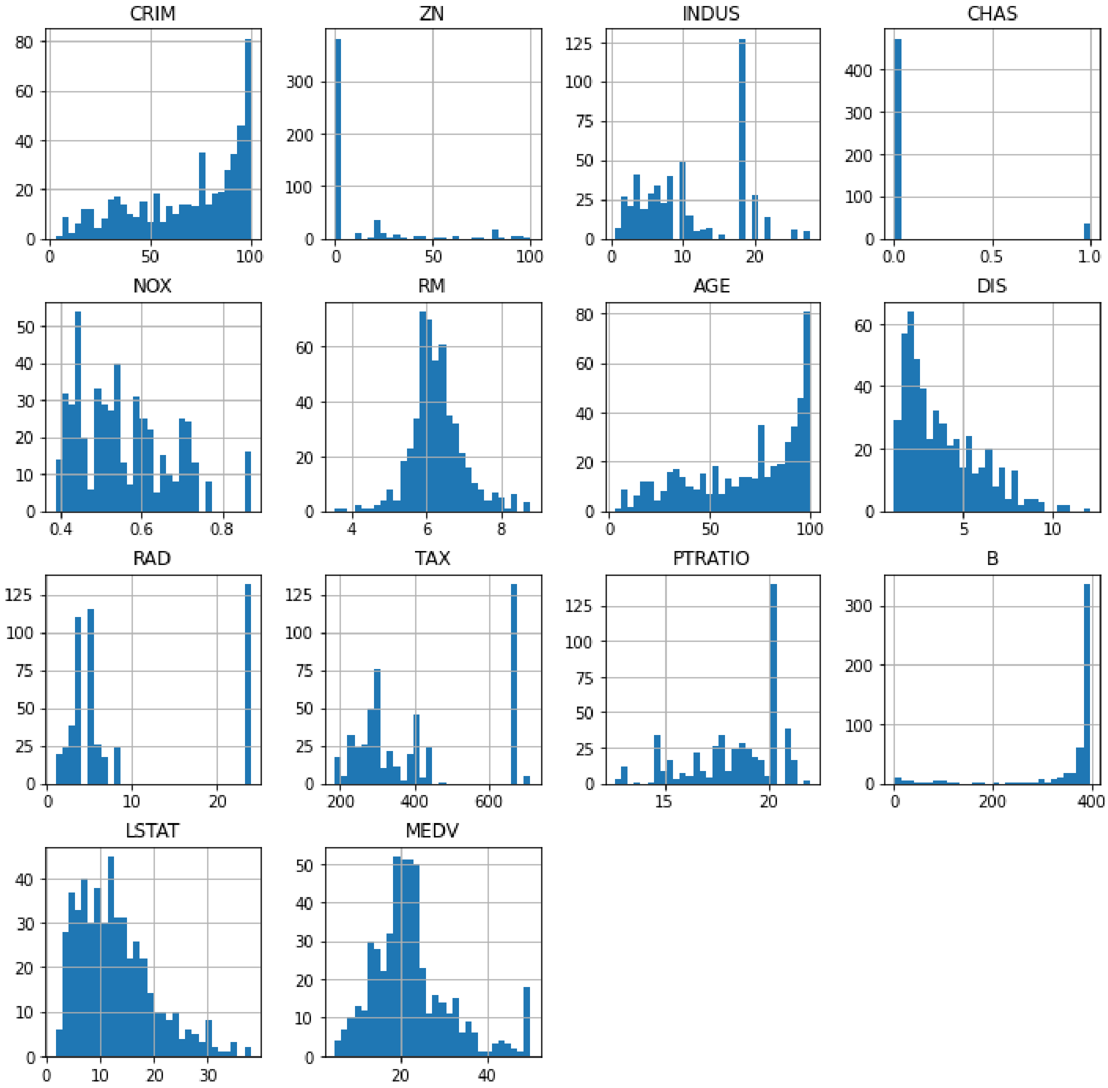
[6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CRIM     506 non-null    float64
 1   ZN       506 non-null    float64
 2   INDUS    506 non-null    float64
 3   CHAS     506 non-null    float64
 4   NOX      506 non-null    float64
 5   RM       506 non-null    float64
 6   AGE      506 non-null    float64
 7   DIS      506 non-null    float64
 8   RAD      506 non-null    int64
 9   TAX      506 non-null    int64
 10  PTRATIO  506 non-null    float64
 11  B        506 non-null    float64
 12  LSTAT    506 non-null    float64
 13  MEDV     506 non-null    float64
dtypes: float64(12), int64(2)
memory usage: 55.5 KB
```

[7]:

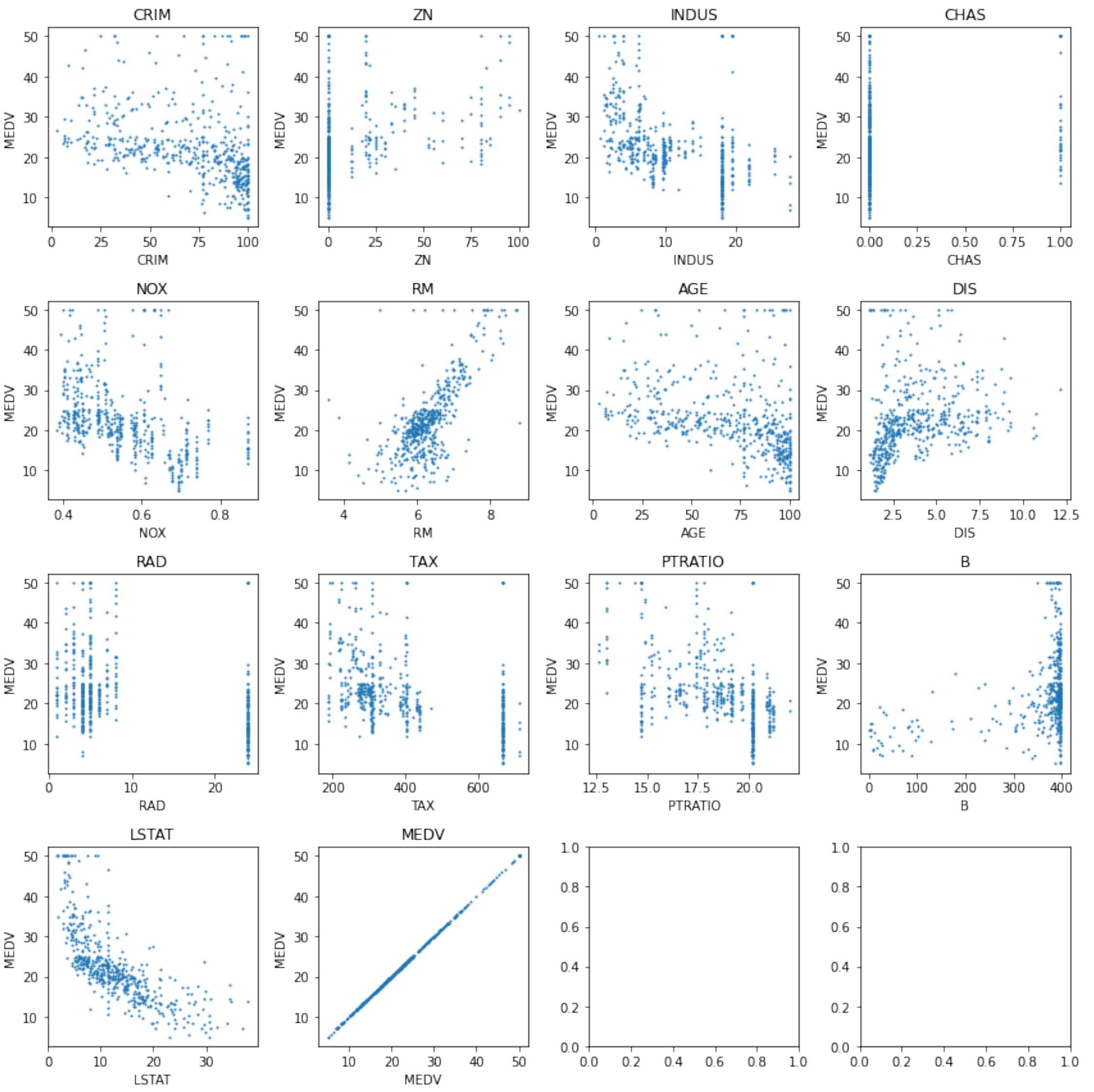
```
_ = df.hist(bins=30, figsize=(12,12))
```



[8]:

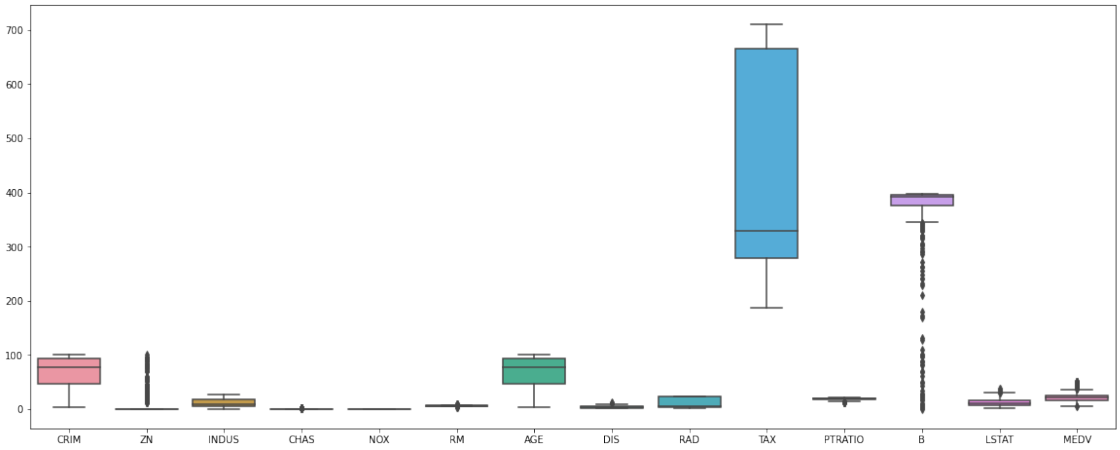
```
# Let's plot against the target label
import matplotlib.pyplot as plt
fig, axs = plt.subplots(4,4, figsize=(12,12),tight_layout=True)

for i, value in enumerate(df.columns):
    axs.flat[i].scatter(df[value], df["MEDV"], s=1)
    axs.flat[i].set_title(value)
    axs.flat[i].set_ylabel("MEDV")
    axs.flat[i].set_xlabel(value)
```



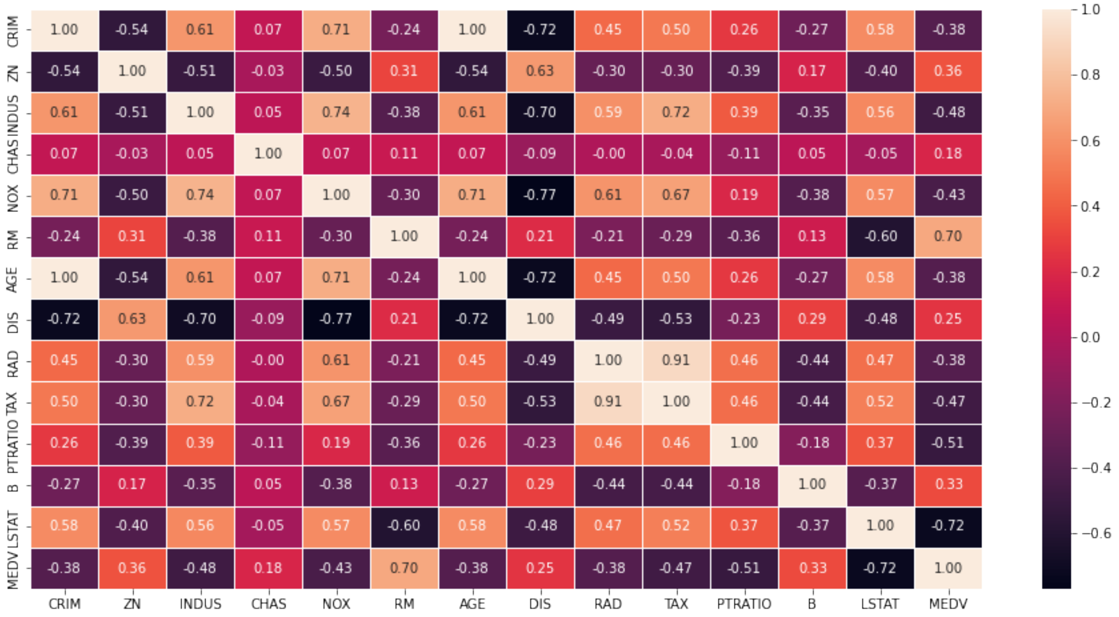
[9]:

```
# looking for outliers using box plot
import seaborn as sns
plt.figure(figsize = (20, 8))
sns.boxplot(data = df, width = 0.8)
plt.show()
```



[10]:
```python
# Heatmap

fig, ax = plt.subplots(figsize = (16, 8))
sns.heatmap(df.corr(), annot = True, fmt = '1.2f', annot_kws = {'size' : 10}, linewidth = 1)
plt.show()
```



[11]:
```python
# From the heatmap we can conclude that we can drop the one of TAX or RAD because they are 0.91 correlated
df.drop("TAX", axis=1, inplace=True)
```

[12]:
```
# For the outliers, lets apply a StandardScalar
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
```

[13]:
```
y = df["MEDV"]
```

[14]:
```
X = df.drop("MEDV", axis=1)
```

[15]:
```
X = scaler.fit_transform(X)
```

[16]:
```
# let's create train and test dataset
```

[17]:
```
from sklearn.model_selection import train_test_split
```

[18]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,␣
 ↪random_state=42)
```

1 Let’s implement linear regression

[19]:
```
from sklearn.linear_model import LinearRegression
```

[20]:
```
lr = LinearRegression()
lr_model = lr.fit(X_train, y_train)
```

[21]:
```
# prediction of model
y_pred = lr.predict(X_test)
```

[22]:
```
# test accuracy of model

lr.score(X_test, y_test)
```

[22]:
```
0.7087827171807681
```

[23]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred)
```

[23]:
```
22.038889576045122
```

2 Let’s implement DT regression

[24]:
```
from sklearn.tree import DecisionTreeRegressor
DT_reg = DecisionTreeRegressor()
DT_model = DT_reg.fit(X_train, y_train)
```

[25]:
```
# prediction of model
y_pred = DT_model.predict(X_test)
```

[26]:
```
# test accuracy of model
DT_model.score(X_test, y_test)
```

[26]:
```
0.6627990849152358
```

[27]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred) # worst error
```

[27]:
```
25.518862275449106
```

3 Let's implement SVR

[28]:
```
from sklearn.svm import SVR
SVR_reg = SVR()
SVR_model = SVR_reg.fit(X_train, y_train)
```

[29]:
```
# prediction of model
y_pred = SVR_model.predict(X_test)
```

[30]:
```
# test accuracy of model
SVR_model.score(X_test, y_test)
```

[30]:
```
0.6692824680855471
```

[31]:
```
from sklearn.metrics import mean_squared_error
mean_squared_error(y_test, y_pred) # worst error
```

[31]:
```
25.028209507912724
```

4 Let's implement Lasso Regression

[32]:
```
from sklearn.linear_model import Lasso, LassoCV
```

[33]:
```
lasso_cv = LassoCV(alphas = None, cv = 10, max_iter = 100000, normalize = True)
```

[34]:
```
lasso_model = lasso_cv.fit(X_train, y_train)
```

```
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Lasso())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * np.sqrt(n_samples).
  warnings.warn(
```

[35]:

```python
# best alpha parameter
alpha = lasso_cv.alpha_
```

[36]:

```python
lasso = Lasso(alpha = lasso_cv.alpha_)
lasso_model = lasso.fit(X_train, y_train)
```

[37]:

```python
lasso.score(X_test, y_test)
```

[37]:

```
0.7085465404392775
```

[38]:

```python
# prediction of model
y_pred = lasso_model.predict(X_test)
mean_squared_error(y_test, y_pred)
```

[38]:

```
22.05676307955342
```

5 Let's Implement Ridge Regression

[39]:

```python
from sklearn.linear_model import Ridge, RidgeCV
import numpy as np
alphas = np.random.uniform(0, 10, 50)
ridge_cv = RidgeCV(alphas = alphas, cv = 10, normalize = True)
ridge_model = ridge_cv.fit(X_train, y_train)
```

```
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())
```

```
If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

```
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), Ridge())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to: original_alpha * n_samples.
  warnings.warn(
```

[40]:

```
ridge_model.score(X_test, y_test)
```

[40]:

```
0.7074456396202318
```

[41]:

```
# prediction of model
y_pred = ridge_model.predict(X_test)
mean_squared_error(y_test, y_pred)
```

[41]:

```
22.140077611404827
```

6 Let's implement ElasticNet

[42]:
```
from sklearn.linear_model import ElasticNet, ElasticNetCV
elastic_net_cv = ElasticNetCV(alphas = None, cv = 10, max_iter = 100000,␣
 ↪normalize = True)
elastic_model = elastic_net_cv.fit(X_train, y_train)
```

```
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

from sklearn.pipeline import make_pipeline

model = make_pipeline(StandardScaler(with_mean=False), ElasticNet())

If you wish to pass a sample_weight parameter, you need to pass it as a fit
parameter to each step of the pipeline as follows:

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

Set parameter alpha to original_alpha * np.sqrt(n_samples) if l1_ratio is 1, and
to original_alpha * n_samples if l1_ratio is 0. For other values of l1_ratio, no
analytic formula is available.
  warnings.warn(
```

[43]:
```
# l1 ratio
elastic_net_cv.l1_ratio
```

[43]:
```
0.5
```

[44]:
```
elastic_net = ElasticNet(alpha = elastic_net_cv.alpha_, l1_ratio =␣
 ↪elastic_net_cv.l1_ratio)
elastic_model = elastic_net.fit(X_train, y_train)
```

```
/home/local/FARFETCH/tiago.cabo/anaconda3/envs/EDIT/lib/python3.9/site-
packages/sklearn/linear_model/_coordinate_descent.py:648: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 1.530e+03, tolerance: 2.988e+00
  model = cd_fast.enet_coordinate_descent(
```

[45]:
```
elastic_model.score(X_test, y_test)
```

[45]:
```
0.7087696148075188
```

[46]:

```
# prediction of model
y_pred = elastic_model.predict(X_test)
mean_squared_error(y_test, y_pred)
```

[46]:

```
22.039881144108776
```

[ ]:

```

```