While ridge regularization uses an L2 norm penalty term, another regularization method called LASSO uses an L1 norm for the weights penalty term. Specifically, LASSO regularization will find the optimal weights to minimize the following quantity:

$$\alpha||w||_1 + \sum_{i=1}^{n}(\mathbf{x}_i \cdot w - y_i)^2$$

where $||w||_1$ represents the L1 norm of the weights.

LASSO regularization tends to prefer linear models with fewer parameter values. This means that it will likely zero-out some of the weight coefficients. This reduces the number of features that the model is actually dependent on (since some of the coefficients will now be 0), which can be beneficial when some features are completely irrelevant or duplicates of other features.

In scikit-learn, we implement LASSO using the `Lasso` object, which is part of the `linear_model` module. Like the `Ridge` object, it takes in the model's α value with the `alpha` keyword argument (default is 1.0).

The code below demonstrates how to use the `Lasso` object on a dataset with 150 observations and 4 features.

```python
# predefined dataset
print('Data shape: {}\n'.format(data.shape))
print('Labels shape: {}\n'.format(labels.shape))

from sklearn import linear_model
reg = linear_model.Lasso(alpha=0.1)
reg.fit(data, labels)
print('Coefficients: {}\n'.format(repr(reg.coef_)))
print('Intercept: {}\n'.format(reg.intercept_))
print('R2: {}\n'.format(reg.score(data, labels)))
```

RUN SAVE RESET

Close

Output 1.300s

```
Data shape: (150, 4)

Labels shape: (150,)

Coefficients: array([ 0.        , -0.        ,  0.40830957,  0.        ])

Intercept: -0.534699558318563

R2: 0.895831189504504
```

In the example above, note that a majority of the weights are 0, due to the LASSO sparse weight preference.

There is also a cross-validated version in the form of the `LassoCV` object, which works in essentially the same way as the `RidgeCV` object.