# **[sklearn.linear\_model](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.linear_model" \o "sklearn.linear_model).LinearRegression**

*class*sklearn.linear\_model.**LinearRegression**(*fit\_intercept=True*, *normalize=False*, *copy\_X=True*, *n\_jobs=1*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/ef5cb84a/sklearn/linear_model/base.py" \l "L405)

Ordinary least squares Linear Regression.

|  |  |
| --- | --- |
| **Parameters:** | ****fit\_intercept**** : boolean, optional, default True一个布尔值，指定是否需要计算b的值，如果为False，那么不计算b值。  whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (e.g. data is expected to be already centered).  ****normalize**** : boolean, optional, default False一个布尔值，如果为true那么训练样本会在回归之前会被归一化。  This parameter is ignored when fit\_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use **[sklearn.preprocessing.StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html" \l "sklearn.preprocessing.StandardScaler" \o "sklearn.preprocessing.StandardScaler)** before calling fit on an estimator with normalize=False.  ****copy\_X**** : boolean, optional, default True 一个布尔值，如果为True，则会赋值X  If True, X will be copied; else, it may be overwritten.  ****n\_jobs**** : int, optional, default 1一个正数。任务并行时指定的CPU数量，如果为-1则使用所有可用的cpu.  The number of jobs to use for the computation. If -1 all CPUs are used. This will only provide speedup for n\_targets > 1 and sufficient large problems. |
| **Attributes:** | ****coef\_**** : array, shape (n\_features, ) or (n\_targets, n\_features)权重向量  Estimated coefficients for the linear regression problem. If multiple targets are passed during the fit (y 2D), this is a 2D array of shape (n\_targets, n\_features), while if only one target is passed, this is a 1D array of length n\_features.  ****intercept\_**** : array b的值  Independent term in the linear model. |
|  |  |

**Methods**

|  |  |
| --- | --- |
| **[fit](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression.fit" \o "sklearn.linear_model.LinearRegression.fit)**(X, y[, sample\_weight]) | Fit linear model.训练线性回归模型 |
| **[get\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression.get_params" \o "sklearn.linear_model.LinearRegression.get_params)**([deep]) | Get parameters for this estimator. |
| **[predict](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression.predict" \o "sklearn.linear_model.LinearRegression.predict)**(X) | Predict using the linear model用于模型的预测，返回预测值 |
| **[score](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression.score" \o "sklearn.linear_model.LinearRegression.score)**(X, y[, sample\_weight]) | Returns the coefficient of determination R^2 of the prediction.  返回预测性能得分，设预测集为T，真实值为yi，真实值的均值为，则    Score不超过1，可能为负值，效果很差  Score越大，预测性能越好 |
| **[set\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression.set_params" \o "sklearn.linear_model.LinearRegression.set_params)**(\*\*params) | Set the parameters of this estimator. |

岭回归：

# **[sklearn.linear\_model](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.linear_model" \o "sklearn.linear_model).Ridge 岭回归**

*class*sklearn.linear\_model.**Ridge**(*alpha=1.0*, *fit\_intercept=True*, *normalize=False*, *copy\_X=True*, *max\_iter=None*, *tol=0.001*, *solver=’auto’*, *random\_state=None*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/ef5cb84a/sklearn/linear_model/ridge.py" \l "L512)

Linear least squares with l2 regularization.**加入L2范数惩罚项的线性回归**

This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression (i.e., when y is a 2d-array of shape [n\_samples, n\_targets]).

Read more in the [User Guide](http://scikit-learn.org/stable/modules/linear_model.html" \l "ridge-regression).

|  |  |
| --- | --- |
| **Parameters:** | ****alpha**** : {float, array-like}, shape (n\_targets)其值越大正则化项的占比越大  Regularization strength; must be a positive float. Regularization improves the conditioning of the problem and reduces the variance of the estimates. Larger values specify stronger regularization. Alpha corresponds to C^-1 in other linear models such as LogisticRegression or LinearSVC. If an array is passed, penalties are assumed to be specific to the targets. Hence they must correspond in number.  ****fit\_intercept**** : boolean一个布尔值，指定是否需要计算b的值。如果为false，那么不计算b值（模型会假设你的数据已经中心化了）  Whether to calculate the intercept for this model. If set to false, no intercept will be used in calculations (e.g. data is expected to be already centered).  ****normalize**** : boolean, optional, default False一个布尔值，如果为true，那么训练样本会在回归之前被归一化  This parameter is ignored when fit\_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use **[sklearn.preprocessing.StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html" \l "sklearn.preprocessing.StandardScaler" \o "sklearn.preprocessing.StandardScaler)** before calling fit on an estimator with normalize=False.  ****copy\_X**** : boolean, optional, default True布尔值，为true，则会复制X  If True, X will be copied; else, it may be overwritten.  ****max\_iter**** : int, optional整数，最大迭代次数  Maximum number of iterations for conjugate gradient solver. For ‘sparse\_cg’ and ‘lsqr’ solvers, the default value is determined by scipy.sparse.linalg. For ‘sag’ solver, the default value is 1000.  ****tol**** : float一个浮点数，指定判定迭代迭代收敛与否的阈值  Precision of the solution.  ****solver**** : {‘auto’, ‘svd’, ‘cholesky’, ‘lsqr’, ‘sparse\_cg’, ‘sag’, ‘saga’}  Solver to use in the computational routines:  指定求解最优解问题的算法，下列是可选值   * ‘auto’ chooses the solver automatically based on the type of data. * 根据数据集自动选择算法 * ‘svd’ uses a Singular Value Decomposition of X to compute the Ridge coefficients. More stable for singular matrices than ‘cholesky’. * 使用奇异值分解来计算回归系数 * ‘cholesky’ uses the standard scipy.linalg.solve function to obtain a closed-form solution. * 使用scipy.linalg.solve函数来求解 * ‘sparse\_cg’ uses the conjugate gradient solver as found in scipy.sparse.linalg.cg. As an iterative algorithm, this solver is more appropriate than ‘cholesky’ for large-scale data (possibility to set tol and max\_iter).使用scipy.sparse.linalg求解，速度最快 * ‘lsqr’ uses the dedicated regularized least-squares routine scipy.sparse.linalg.lsqr. It is the fastest but may not be available in old scipy versions. It also uses an iterative procedure. * ‘sag’ uses a Stochastic Average Gradient descent, and ‘saga’ uses its improved, unbiased version named SAGA. Both methods also use an iterative procedure, and are often faster than other solvers when both n\_samples and n\_features are large. Note that ‘sag’ and ‘saga’ fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from sklearn.preprocessing. * 使用随机平均梯度下降法求解最优化问题。   All last five solvers support both dense and sparse data. However, only ‘sag’ and ‘saga’ supports sparse input when fit\_intercept is True.所有最后五个求解器都支持密集和稀疏数据。 但是，当fit\_intercept为True时，只有'sag'和'saga'支持稀疏输入。  ***New in version 0.17****:*Stochastic Average Gradient descent solver.  *New in version 0.19:*SAGA solver.  ****random\_state**** : int, RandomState instance or None, optional, default None  The seed of the pseudo random number generator to use when shuffling the data. If int, random\_state is the seed used by the random number generator; If RandomState instance, random\_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random. Used when solver == ‘sag’.  如果为整数，则它值定了随机数生成器的种子  如果为RandomState 实例，则制定可随机数生成器  如果为None，则使用默认随机数生成器  *New in version 0.17: random\_state* to support Stochastic Average Gradient. |
| **Attributes:** | ****coef\_**** :权重向量 array, shape (n\_features,) or (n\_targets, n\_features)  Weight vector(s).  ****intercept\_**** : b值--float | array, shape = (n\_targets,)  Independent term in decision function. Set to 0.0 if fit\_intercept = False.  ****n\_iter\_**** :实际迭代次数 array or None, shape (n\_targets,)  Actual number of iterations for each target. Available only for sag and lsqr solvers. Other solvers will return None.  *New in version 0.17.* |

**Methods**

|  |  |
| --- | --- |
| **[fit](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html" \l "sklearn.linear_model.Ridge.fit" \o "sklearn.linear_model.Ridge.fit)**(X, y[, sample\_weight]) | Fit Ridge regression model 训练岭回归线性模型 |
| **[get\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html" \l "sklearn.linear_model.Ridge.get_params" \o "sklearn.linear_model.Ridge.get_params)**([deep]) | Get parameters for this estimator. |
| **[predict](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html" \l "sklearn.linear_model.Ridge.predict" \o "sklearn.linear_model.Ridge.predict)**(X) | Predict using the linear model用模型进行预测，返回预测值 |
| **[score](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html" \l "sklearn.linear_model.Ridge.score" \o "sklearn.linear_model.Ridge.score)**(X, y[, sample\_weight]) | Returns the coefficient of determination R^2 of the prediction.  返回预测性能得分，和线性回归的公式一样，值越大性能越好 |
| **[set\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html" \l "sklearn.linear_model.Ridge.set_params" \o "sklearn.linear_model.Ridge.set_params)**(\*\*params) | Set the parameters of this estimator. |

Lasso回归

# **[sklearn.linear\_model](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.linear_model" \o "sklearn.linear_model).Lasso 回归**

*class*sklearn.linear\_model.**Lasso**(*alpha=1.0*, *fit\_intercept=True*, *normalize=False*, *precompute=False*, *copy\_X=True*, *max\_iter=1000*, *tol=0.0001*, *warm\_start=False*, *positive=False*, *random\_state=None*, *selection=’cyclic’*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/ef5cb84a/sklearn/linear_model/coordinate_descent.py" \l "L797)

Linear Model trained with L1 prior as regularizer (aka the Lasso)

The optimization objective for Lasso is:

(1 / (2 \* n\_samples)) \* ||y - Xw||^2\_2 + alpha \* ||w||\_1

Technically the Lasso model is optimizing the same objective function as the Elastic Net with l1\_ratio=1.0 (no L2 penalty).

Read more in the [User Guide](http://scikit-learn.org/stable/modules/linear_model.html" \l "lasso).

|  |  |
| --- | --- |
| **Parameters:** | ****alpha**** : float, optional 值越大正则化项占比越大  Constant that multiplies the L1 term. Defaults to 1.0. alpha = 0 is equivalent to an ordinary least square, solved by the **[LinearRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression" \o "sklearn.linear_model.LinearRegression)** object. For numerical reasons, using alpha = 0 with the Lasso object is not advised. Given this, you should use the **[LinearRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression" \o "sklearn.linear_model.LinearRegression)** object.  ****fit\_intercept**** : boolean 一个布尔值，指定时候需要计算b值，如果weiFalse，那么不会计算b值（模型已经假设你已经中心化了）。  whether to calculate the intercept for this model. If set to false, no intercept will be used in calculations (e.g. data is expected to be already centered).  ****normalize**** : boolean, optional, default False如果为True，训练样本会在回归之前被归一化  This parameter is ignored when fit\_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use **[sklearn.preprocessing.StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html" \l "sklearn.preprocessing.StandardScaler" \o "sklearn.preprocessing.StandardScaler)** before calling fit on an estimator with normalize=False.  ****precompute**** : True | False | array-like, default=False一个布尔值或一个序列。他决定了是否提前计算Gram矩阵来加速运算。  Whether to use a precomputed Gram matrix to speed up calculations. If set to 'auto' let us decide. The Gram matrix can also be passed as argument. For sparse input this option is always True to preserve sparsity.  ****copy\_X**** : boolean, optional, default True 如果weiTrue，则会复制X  If True, X will be copied; else, it may be overwritten.  ****max\_iter**** : int, optional最大迭代次数  The maximum number of iterations  ****tol**** : float, optional指定判断迭代收敛与否的阈值  The tolerance for the optimization: if the updates are smaller than tol, the optimization code checks the dual gap for optimality and continues until it is smaller than tol.  ****warm\_start**** : bool, optional一个布尔值，如果为true，那么使用前一次训练结果继续训练。否则从头开始训练  When set to True, reuse the solution of the previous call to fit as initialization, otherwise, just erase the previous solution.  ****positive**** : bool, optional一个布尔值，如果为true，那么强制要求权重向量的分量都是正数。  When set to True, forces the coefficients to be positive.  ****random\_state**** : int, RandomState instance or None, optional, default None  一个整数或者一个Randonstate实例，或者为None  The seed of the pseudo random number generator that selects a random feature to update. If int, random\_state is the seed used by the random number generator; If RandomState instance, random\_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random. Used when selection == ‘random’.  ****selection**** : str, default ‘cyclic’一个字符串，可以取值cyclic或random，指定了每次迭代的时候选择权重向量的那个分量来更新  Random：为整数，则它指定了随机选择权重向量的一个分量来更新  Cyclic：更新的时候，从前往后依次选择权重向量的一个分量来更新  If set to ‘random’, a random coefficient is updated every iteration rather than looping over features sequentially by default. This (setting to ‘random’) often leads to significantly faster convergence especially when tol is higher than 1e-4. |
| **Attributes:** | ****coef\_**** : 权重向量，array, shape (n\_features,) | (n\_targets, n\_features)  parameter vector (w in the cost function formula)  ****sparse\_coef\_**** : 稀疏系数scipy.sparse matrix, shape (n\_features, 1) | (n\_targets, n\_features)  sparse\_coef\_ is a readonly property derived from coef\_  ****intercept\_**** :b的值， float | array, shape (n\_targets,)  independent term in decision function.  ****n\_iter\_**** :实际迭代次数 int | array-like, shape (n\_targets,)  number of iterations run by the coordinate descent solver to reach the specified tolerance. |

**Methods**

|  |  |
| --- | --- |
| **[fit](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html" \l "sklearn.linear_model.Lasso.fit" \o "sklearn.linear_model.Lasso.fit)**(X, y[, check\_input]) | Fit model with coordinate descent.训练模型 |
| **[get\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html" \l "sklearn.linear_model.Lasso.get_params" \o "sklearn.linear_model.Lasso.get_params)**([deep]) | Get parameters for this estimator. |
| **[path](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html" \l "sklearn.linear_model.Lasso.path" \o "sklearn.linear_model.Lasso.path)**(X, y[, l1\_ratio, eps, n\_alphas, …]) | Compute elastic net path with coordinate descent计算具有坐标下降的弹性网路径 |
| **[predict](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html" \l "sklearn.linear_model.Lasso.predict" \o "sklearn.linear_model.Lasso.predict)**(X) | Predict using the linear model 用模型进行预测 |
| **[score](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html" \l "sklearn.linear_model.Lasso.score" \o "sklearn.linear_model.Lasso.score)**(X, y[, sample\_weight]) | Returns the coefficient of determination R^2 of the prediction.返回预测性能得分 |
| **[set\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html" \l "sklearn.linear_model.Lasso.set_params" \o "sklearn.linear_model.Lasso.set_params)**(\*\*params) | Set the parameters of this estimator. |

弹性网络回归

# **[sklearn.linear\_model](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.linear_model" \o "sklearn.linear_model).ElasticNet 回归**

*class*sklearn.linear\_model.**ElasticNet**(*alpha=1.0*, *l1\_ratio=0.5*, *fit\_intercept=True*, *normalize=False*, *precompute=False*, *max\_iter=1000*, *copy\_X=True*, *tol=0.0001*, *warm\_start=False*, *positive=False*, *random\_state=None*, *selection=’cyclic’*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/ef5cb84a/sklearn/linear_model/coordinate_descent.py" \l "L510)

Linear regression with combined L1 and L2 priors as regularizer.

Minimizes the objective function:

1 / (2 \* n\_samples) \* ||y - Xw||^2\_2+ alpha \* l1\_ratio \* ||w||\_1+ 0.5 \* alpha \* (1 - l1\_ratio) \* ||w||^2\_2

If you are interested in controlling the L1 and L2 penalty separately, keep in mind that this is equivalent to:

a \* L1 + b \* L2

where:

alpha = a + b **and** l1\_ratio = a / (a + b)

The parameter l1\_ratio corresponds to alpha in the glmnet R package while alpha corresponds to the lambda parameter in glmnet. Specifically, l1\_ratio = 1 is the lasso penalty. Currently, l1\_ratio <= 0.01 is not reliable, unless you supply your own sequence of alpha.

Read more in the [User Guide](http://scikit-learn.org/stable/modules/linear_model.html" \l "elastic-net).

|  |  |
| --- | --- |
| **Parameters:** | ****alpha**** : float, optional**增加惩罚条件的常数。** 默认为1.0。请参阅这个参数的确切数学含义的说明.``alpha = 0``相当于一个普通的最小二乘法，由LinearRegression对象求解。 出于数值原因，不建议使用alpha = 0与Lasso对象。 给定这个，你应该使用LinearRegression对象。  Constant that multiplies the penalty terms. Defaults to 1.0. See the notes for the exact mathematical meaning of this parameter.``alpha = 0`` is equivalent to an ordinary least square, solved by the **[LinearRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression" \o "sklearn.linear_model.LinearRegression)** object. For numerical reasons, using alpha = 0 with the Lasso object is not advised. Given this, you should use the **[LinearRegression](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html" \l "sklearn.linear_model.LinearRegression" \o "sklearn.linear_model.LinearRegression)** object.  ****l1\_ratio**** : float--ElasticNet混合参数，0 <= l1\_ratio <= 1。对于l1\_ratio = 0，惩罚是L2惩罚。 对于l1\_ratio = 1，它是一个L1惩罚。 对于0 <l1\_ratio <1，惩罚是L1和L2的组合。  The ElasticNet mixing parameter, with 0 <= l1\_ratio <= 1. For l1\_ratio = 0 the penalty is an L2 penalty. For l1\_ratio = 1 it is an L1 penalty. For 0 < l1\_ratio < 1, the penalty is a combination of L1 and L2.  ****fit\_intercept**** : bool是否应该估计截距b的值。 如果False，假定数据已经居中。  Whether the intercept should be estimated or not. If False, the data is assumed to be already centered.  ****normalize**** : boolean, optional, default False一个布尔值，如果为True那么训练样本会在回归之前被归一化。  This parameter is ignored when fit\_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use **[sklearn.preprocessing.StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html" \l "sklearn.preprocessing.StandardScaler" \o "sklearn.preprocessing.StandardScaler)** before calling fit on an estimator with normalize=False.  ****precompute**** : True | False | array-like 布尔值或序列，决定了是否提前计算Gram矩阵来加速计算  Whether to use a precomputed Gram matrix to speed up calculations. The Gram matrix can also be passed as argument. For sparse input this option is always True to preserve sparsity.  ****max\_iter**** : int, optional最大迭代次数  The maximum number of iterations  ****copy\_X**** : boolean, optional, default True 为True复制x  If True, X will be copied; else, it may be overwritten.  ****tol**** : float, optional指定判断迭代收敛与否的阈值  The tolerance for the optimization: if the updates are smaller than tol, the optimization code checks the dual gap for optimality and continues until it is smaller than tol.  ****warm\_start**** : bool, optional如果为true，那么使用前一次训练结果继续训练，否则从头开始训练  When set to True, reuse the solution of the previous call to fit as initialization, otherwise, just erase the previous solution.  ****positive**** : bool, optional 如果为True，那么强制要求权重向量的分量都为正数  When set to True, forces the coefficients to be positive.  ****random\_state**** : int, RandomState instance or None, optional, default None  The seed of the pseudo random number generator that selects a random feature to update. If int, random\_state is the seed used by the random number generator; If RandomState instance, random\_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random. Used when selection == ‘random’.  ****selection**** : str, default ‘cyclic’一个字符串，取值为cyclic’或random  Random更新的时候，随机选择权重向量的一个分量过来更新  cyclic’更新的时候，从前往后依次选择权重向量的一个分量来更新  If set to ‘random’, a random coefficient is updated every iteration rather than looping over features sequentially by default. This (setting to ‘random’) often leads to significantly faster convergence especially when tol is higher than 1e-4. |
| **Attributes:** | ****coef\_**** : 权重向量array, shape (n\_features,) | (n\_targets, n\_features)  parameter vector (w in the cost function formula)  ****sparse\_coef\_**** : scipy.sparse matrix, shape (n\_features, 1) | (n\_targets, n\_features)  sparse\_coef\_ is a readonly property derived from coef\_  ****intercept\_**** :b值 float | array, shape (n\_targets,)  independent term in decision function.  ****n\_iter\_**** : 实际迭代次数array-like, shape (n\_targets,)  number of iterations run by the coordinate descent solver to reach the specified tolerance. |

**Methods**

|  |  |
| --- | --- |
| **[fit](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html" \l "sklearn.linear_model.ElasticNet.fit" \o "sklearn.linear_model.ElasticNet.fit)**(X, y[, check\_input]) | Fit model with coordinate descent.训练模型 |
| **[get\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html" \l "sklearn.linear_model.ElasticNet.get_params" \o "sklearn.linear_model.ElasticNet.get_params)**([deep]) | Get parameters for this estimator. |
| **[path](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html" \l "sklearn.linear_model.ElasticNet.path" \o "sklearn.linear_model.ElasticNet.path)**(X, y[, l1\_ratio, eps, n\_alphas, …]) | Compute elastic net path with coordinate descent |
| **[predict](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html" \l "sklearn.linear_model.ElasticNet.predict" \o "sklearn.linear_model.ElasticNet.predict)**(X) | Predict using the linear model模型预测，返回预测值 |
| **[score](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html" \l "sklearn.linear_model.ElasticNet.score" \o "sklearn.linear_model.ElasticNet.score)**(X, y[, sample\_weight]) | Returns the coefficient of determination R^2 of the prediction.返回预测性能得分 |
| **[set\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html" \l "sklearn.linear_model.ElasticNet.set_params" \o "sklearn.linear_model.ElasticNet.set_params)**(\*\*params) | Set the parameters of this estimator. |

逻辑斯特回归

# **[sklearn.linear\_model](http://scikit-learn.org/stable/modules/classes.html" \l "module-sklearn.linear_model" \o "sklearn.linear_model).LogisticRegression逻辑斯特回归**

*class*sklearn.linear\_model.**LogisticRegression**(*penalty=’l2’*, *dual=False*, *tol=0.0001*, *C=1.0*, *fit\_intercept=True*, *intercept\_scaling=1*, *class\_weight=None*, *random\_state=None*, *solver=’liblinear’*, *max\_iter=100*, *multi\_class=’ovr’*, *verbose=0*, *warm\_start=False*, *n\_jobs=1*)[[source]](https://github.com/scikit-learn/scikit-learn/blob/ef5cb84a/sklearn/linear_model/logistic.py" \l "L962)

Logistic Regression (aka logit, MaxEnt) classifier.

In the multiclass case, the training algorithm uses the one-vs-rest (OvR) scheme if the ‘multi\_class’ option is set to ‘ovr’, and uses the cross- entropy loss if the ‘multi\_class’ option is set to ‘multinomial’. (Currently the ‘multinomial’ option is supported only by the ‘lbfgs’, ‘sag’ and ‘newton-cg’ solvers.)

This class implements regularized logistic regression using the ‘liblinear’ library, ‘newton-cg’, ‘sag’ and ‘lbfgs’ solvers. It can handle both dense and sparse input. Use C-ordered arrays or CSR matrices containing 64-bit floats for optimal performance; any other input format will be converted (and copied).

The ‘newton-cg’, ‘sag’, and ‘lbfgs’ solvers support only L2 regularization with primal formulation. The ‘liblinear’ solver supports both L1 and L2 regularization, with a dual formulation only for the L2 penalty.

Read more in the [User Guide](http://scikit-learn.org/stable/modules/linear_model.html" \l "logistic-regression).

|  |  |
| --- | --- |
| **Parameters:** | ****penalty**** : str, ‘l1’ or ‘l2’, default: ‘l2’字符出指定了正则化策略  Used to specify the norm used in the penalization. The ‘newton-cg’, ‘sag’ and ‘lbfgs’ solvers support only l2 penalties.  *New in version 0.19:*l1 penalty with SAGA solver (allowing ‘multinomial’ + L1)  ****dual**** : bool, default: False一个布尔值，如果为true则求解对偶形式（在penalty=l2且solver=linlinear才有对偶形式），如果为false求解原始形式  Dual or primal formulation. Dual formulation is only implemented for l2 penalty with liblinear solver. Prefer dual=False when n\_samples > n\_features.  ****tol**** : float, default: 1e-4迭代收敛与否的阈值，默认1e-4  Tolerance for stopping criteria.  ****C**** : float, default: 1.0一个浮点数，指定了罚项系数的倒数。值越小正则化项越大  Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.  ****fit\_intercept**** : bool, default: True指定是否需要b的值，为false不会计算b的值  Specifies if a constant (a.k.a. bias or intercept) should be added to the decision function.  ****intercept\_scaling**** : float, default 1.只有当solver=liblinear才有意义。当采用fit\_intercept时，相当于人造一个特征出来，该特征恒为1，其权重为b。在计算正则化项的时候，改人造特征也被考虑了。因此为了降低人造特征的影响，需要提供****intercept\_scaling****  Useful only when the solver ‘liblinear’ is used and self.fit\_intercept is set to True. In this case, x becomes [x, self.intercept\_scaling], i.e. a “synthetic” feature with constant value equal to intercept\_scaling is appended to the instance vector. The intercept becomes intercept\_scaling \* synthetic\_feature\_weight.  Note! the synthetic feature weight is subject to l1/l2 regularization as all other features. To lessen the effect of regularization on synthetic feature weight (and therefore on the intercept) intercept\_scaling has to be increased.  ****class\_weight**** : dict or ‘balanced’, default: None权重  Balance每个分类的权重与该分类在样本集中出现频率成反比。为None则每个权重都为1，为字典，分别给出每个分类的权重  Weights associated with classes in the form {class\_label: weight}. If not given, all classes are supposed to have weight one.  The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n\_samples / (n\_classes \* np.bincount(y)).  Note that these weights will be multiplied with sample\_weight (passed through the fit method) if sample\_weight is specified.  *New in version 0.17: class\_weight=’balanced’*  ****random\_state**** : int, RandomState instance or None, optional, default: None  The seed of the pseudo random number generator to use when shuffling the data. If int, random\_state is the seed used by the random number generator; If RandomState instance, random\_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random. Used when solver == ‘sag’ or ‘liblinear’.  ****solver**** : {‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’}适用的最优化问题的算法  **newton-cg牛顿法**  **Lbfgs 使用L-BFG拟牛顿法**  **Liblinear：使用liblinear**  **Sag：使用随机平均梯度下降法**  **对于规模较小的数据集，liblinear比较适用，大规模数据集sag适用**  **Newton-cg、lbfgs、sag只处理penalty=l2的情况。**  default: ‘liblinear’ Algorithm to use in the optimization problem.  **For small datasets, ‘liblinear’ is a good choice, whereas ‘sag’ and**  ‘saga’ are faster for large ones.  **For multiclass problems, only ‘newton-cg’, ‘sag’, ‘saga’ and ‘lbfgs’**  handle multinomial loss; ‘liblinear’ is limited to one-versus-rest schemes.  **‘newton-cg’, ‘lbfgs’ and ‘sag’ only handle L2 penalty, whereas**  ‘liblinear’ and ‘saga’ handle L1 penalty.  Note that ‘sag’ and ‘saga’ fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from sklearn.preprocessing.  *New in version 0.17:*Stochastic Average Gradient descent solver.  *New in version 0.19:*SAGA solver.  ****max\_iter**** : int, default: 100 最大迭代次数  Useful only for the newton-cg, sag and lbfgs solvers. Maximum number of iterations taken for the solvers to converge.  ****multi\_class**** : str, {‘ovr’, ‘multinomial’}, default: ‘ovr’  指定多分类问题策略  **Ovr-one-vs-rest策略**  **Multinomial是直接采用多分类逻辑回归策略**  Multiclass option can be either ‘ovr’ or ‘multinomial’. If the option chosen is ‘ovr’, then a binary problem is fit for each label. Else the loss minimised is the multinomial loss fit across the entire probability distribution. Does not work for liblinear solver.  *New in version 0.18:*Stochastic Average Gradient descent solver for ‘multinomial’ case.  ****verbose**** : int, default: 0  For the liblinear and lbfgs solvers set verbose to any positive number for verbosity.  ****warm\_start**** : bool, default: False为true那么使用前一次的结果继续训练，否则从头开始。  When set to True, reuse the solution of the previous call to fit as initialization, otherwise, just erase the previous solution. Useless for liblinear solver.  *New in version 0.17: warm\_start* to support *lbfgs*, *newton-cg*, *sag*, *saga* solvers.  ****n\_jobs**** : int, default: 1  Number of CPU cores used when parallelizing over classes if multi\_class=’ovr’”. This parameter is ignored when the [``](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "id1)solver``is set to ‘liblinear’ regardless of whether ‘multi\_class’ is specified or not. If given a value of -1, all cores are used. |
| **Attributes:** | ****coef\_**** : 权重向量array, shape (1, n\_features) or (n\_classes, n\_features)  Coefficient of the features in the decision function.  coef\_ is of shape (1, n\_features) when the given problem is binary.  ****intercept\_**** :b的值 array, shape (1,) or (n\_classes,)  Intercept (a.k.a. bias) added to the decision function.  If fit\_intercept is set to False, the intercept is set to zero. intercept\_ is of shape(1,) when the problem is binary.  ****n\_iter\_**** : 实际迭代次数array, shape (n\_classes,) or (1, )  Actual number of iterations for all classes. If binary or multinomial, it returns only 1 element. For liblinear solver, only the maximum number of iteration across all classes is given. |

**Methods**

|  |  |
| --- | --- |
| **[decision\_function](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.decision_function" \o "sklearn.linear_model.LogisticRegression.decision_function)**(X) | Predict confidence scores for samples. |
| **[densify](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.densify" \o "sklearn.linear_model.LogisticRegression.densify)**() | Convert coefficient matrix to dense array format. |
| **[fit](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.fit" \o "sklearn.linear_model.LogisticRegression.fit)**(X, y[, sample\_weight]) | Fit the model according to the given training data.训练模型 |
| **[get\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.get_params" \o "sklearn.linear_model.LogisticRegression.get_params)**([deep]) | Get parameters for this estimator. |
| **[predict](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.predict" \o "sklearn.linear_model.LogisticRegression.predict)**(X) | Predict class labels for samples in X. |
| **[predict\_log\_proba](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.predict_log_proba" \o "sklearn.linear_model.LogisticRegression.predict_log_proba)**(X) | Log of probability estimates.返回一个数组，数组的元素依次是x预测为各类概率的对数值 |
| **[predict\_proba](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.predict_proba" \o "sklearn.linear_model.LogisticRegression.predict_proba)**(X) | Probability estimates.返回一个数组，数组元素依次是X预测为哥哥类比饿的概率值 |
| **[score](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.score" \o "sklearn.linear_model.LogisticRegression.score)**(X, y[, sample\_weight]) | Returns the mean accuracy on the given test data and labels.返回在x和y上的准确率 |
| **[set\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.set_params" \o "sklearn.linear_model.LogisticRegression.set_params)**(\*\*params) | Set the parameters of this estimator. |
| **[sparsify](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html" \l "sklearn.linear_model.LogisticRegression.sparsify" \o "sklearn.linear_model.LogisticRegression.sparsify)**() | Convert coefficient matrix to sparse format. |