SKLEARN API

**LINEAR REGRESSION:**

Linear Regression fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

It’s parameters are:

* fit\_intercept     bool, default=True

Whether to calculate the intercept for this model. If False then no intercept is used for calculation.

      - copy\_X    bool, default=True

If True, X will be copied; else, it may be overwritten

      - n\_jobs    int, default=None

The number of jobs to use for the computation. This will only provide speedup for n\_targets > 1 and sufficient large problems.

      - positive     bool, default=False

When set to True, forces the coefficients to be positive. This option is only supported for dense arrays.

**LinearRegression(\*, fit\_intercept=True, normalize=False, copy\_X=True, n\_jobs=None, positive=False)**

**Attribute:**

coef\_array of shape (n\_features, )   or                                     Estimated coefficients for  the       (n\_targets, n\_features)      linear regression problem.

rank\_int  Rank of matrix X. Only available when X is dense.

singular\_array of shape (min(X, y),)                              Singular values of X. Only available when X     .                             is dense.

intercept\_float or array of shape (n\_targets,) Independent term in the linear model. Set        .              to 0.0 if fit\_intercept = False.

**Methods**

fit(X, y[, sample\_weight])                         Fit linear model.

get\_params([deep])                                   Get parameters for this estimator.

predict(X)                                                     Predict using the linear model.

score(X, y[, sample\_weight])                    Return the coefficient of determination  of the  .              prediction.

set\_params(\*\*params)                              Set the parameters of this estimator

**LOGISTIC REGRESSION:**

It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable.

Fit(X,y)-fit the model according to the given training data

Predict(x)-predict class labels

Score(X,y)-returns mean accuracy on the given test data and label

Code:

**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='lbfgs', max\_iter=100, multi\_class='auto', verbose=0, warm\_start=False, n\_jobs=None, l1\_ratio=None)**

**RIDGE:**

||y - Xw||^2\_2 + alpha \* ||w||^2\_2

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)).

Ridge regression penalizes the model based on the sum of squares of magnitude of the coefficients.

Alpha-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.

It’s parameters are:

     - alpha{float, ndarray of shape (n\_targets,)}, default=1.0

Regularization strength; must be a positive float. Larger values specify stronger regularization

     - fit\_intercept: bool, default=True

Whether to fit the intercept for this model

     - normalize:     bool, default=False

This parameter is ignored when fit\_intercept is set to False.

     - copy\_Xbool, default=True

If True, X will be copied; else, it may be overwritten

     - max\_iterint, default=None

Maximum number of iterations for conjugate gradient solver

     - tolfloat, default=1e-3

Precision of the solution.

     - solver{‘auto’, ‘svd’, ‘cholesky’, ‘lsqr’, ‘sparse\_cg’, ‘sag’, ‘saga’}, default=’auto’

Solver to use in the computational routines. ‘auto’ chooses the solver automatically based on the type of data.

     - random\_stateint, RandomState instance, default=None

Used when solver == ‘sag’ or ‘saga’ to shuffle the data

**Ridge(alpha=1.0, \*, fit\_intercept=True, normalize=False, copy\_X=True, max\_iter=None, tol=0.001, solver='auto', random\_state=None)**

**Methods** :

fit(X, y[, sample\_weight])          Fit Ridge regression model.

get\_params([deep])     Get parameters for this estimator.

predict(X)     Predict using the linear model.

score(X, y[, sample\_weight])                    Return the coefficient of determination  of the prediction.

set\_params(\*\*params)     Set the parameters of this estimator.

**LASSO :**

LASSO regression penalizes the model based on the sum of magnitude of the coefficients.

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

**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')**

Parameters:

- alpha  float, default=1.0

Constant that multiplies the L1 term. Defaults to 1.0. alpha = 0  is equivalent to an ordinary least square, solved by the LinearRegression object. For numerical reasons, using alpha = 0  with the Lasso object is not advised

- fit\_intercept      bool, default=True

Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (i.e. data is expected to be centered).

- Normalize        bool, default=False

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.

- Precompute   bool or array-like of shape (n\_features, n\_features), default=False

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 False to preserve sparsity.

- copy\_X   bool, default=True

If True, X will be copied; else, it may be overwritten.

- max\_iter  int, default=1000

The maximum number of iterations.

- Tol   float, default=1e-4

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, default=False

When set to True, reuse the solution of the previous call to fit as initialization, otherwise, just erases the previous solution..

- Positive   bool, default=False

When set to True, forces the coefficients to be positive.

- random\_stateint, RandomState instance, default=None

The seed of the pseudo random number generator that selects a random feature to update. Used when selection == ‘random’.

Pass an int for reproducible output across multiple function calls.

- selection{‘cyclic’, ‘random’}, default=’cyclic’

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