

Name: Ankita Dasgupta  
Roll No: J014

## SCIKIT LEARN API SUMMARY

### Linear Regression

#### Parameters:

- `fit_intercept`: default=True (if set to false then data is expected to be centred)
- `normalize`: default=False, ignored if `fit_intercept` is set to False, regressors will be normalized before regression by subtracting the mean and dividing by the l2-norm.
- `copy_X`: default=True, X is copied otherwise overwritten.
- `n_jobs`: default=None, The number of jobs to use for the computation. This will only provide speedup for `n_targets > 1` and sufficient large problems, None means 1.
- `positive`: default=False, when set to True it forces the coefficient to be positive. The option is only supported for dense arrays.

#### Attributes:

- `coef_`: array of shape (n\_features, ) or (n\_targets, n\_features) returns the array containing all the theta values calculated for the linear regression
- `rank_`: int, returns rank of matrix X. Only available when X is dense.
- `singular_`: array of shape (min(X, y),) returns singular values of X. Only available when X is dense.
- `intercept_`: float or array of shape (n\_targets,) returns independent term in the linear model. Set to 0.0 if `fit_intercept = False`.

### Logistic Regression

#### Parameters:

- `penalty`: {'l1', 'l2', 'elasticnet', 'none'}, default='l2'
- `dual`: Dual or primal formulation. Dual formulation is only implemented for the l2 penalty with a liblinear solver. Prefer `dual=False` when `n_samples > n_features`.
- `tol`: default=1e-4, tolerance for stopping criteria.
- `C`: float, default=1.0, inverse of regularization strength; must be a positive float.
- `fit_intercept`: default=True (if set to false then data is expected to be centred)
- `intercept_scaling`: default=1
- `class_weight`: dict or 'balanced', default=None
- `random_state`: int, RandomState instance, default=None
- `solver`: {'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}, default='lbfgs'
- `max_iter`: default=100
- `multi_class`: {'auto', 'ovr', 'multinomial'}, default='auto'
- `verbose`: default=0, for the liblinear and lbfgs solvers set verbose to any positive number for verbosity
- `warm_start`: bool, default=False

- `n_jobs`: default=None, The number of jobs to use for the computation. This will only provide speedup for `n_targets > 1` and sufficient large problems, None means 1.
- `l1_ratio`: float, default=None, elasticnet mixing parameter with  $0 \leq \text{l1\_ratio} \leq 1$ . Used only when `penalty = 'elasticnet'`.

#### Attributes:

- `classes_`: a list of classes known as the classifier
- `coef_`: array of shape `(n_features, )` or `(n_targets, n_features)` returns the array containing all the theta values calculated for the linear regression
- `intercept_`: float or array of shape `(n_targets,)` returns independent term in the linear model. Set to 0.0 if `fit_intercept = False`.
- `n_iter`: ndarray of shape `(n_classes,)` or `(1, )`, actual number of iterations for all classes. If binary or multinomial, it returns only 1 element.

### **Ridge**

#### Parameters:

- `alpha`: {float, ndarray of shape `(n_targets,)`}, default=1.0, if an array is passed, penalties are assumed to be specific to the targets. Hence they must correspond in number.
- `fit_intercept`: default=True (if set to false then data is expected to be centred)
- `normalize`: default=False, ignored if `fit_intercept` is set to False, regressors will be normalized before regression by subtracting the mean and dividing by the l2-norm.
- `copy_X`: default=True, X is copied otherwise overwritten.
- `max_iter`: int, default=None number of iterations for gradient descent.
- `tol`: precision of the solution
- `solver`: {'auto', 'svd', 'cholesky', 'lsqr', 'sparse\_cg', 'sag', 'saga'}, default='auto'
- `random_state`: int, RandomState instance, default=None

#### Attributes:

- `coef_`: array of shape `(n_features, )` or `(n_targets, n_features)` returns the array containing all the theta values calculated for the linear regression
- `intercept_`: float or array of shape `(n_targets,)` returns independent term in the linear model. Set to 0.0 if `fit_intercept = False`.
- `n_iter`: ndarray of shape `(n_classes,)` or `(1, )`, actual number of iterations for all classes. If binary or multinomial, it returns only 1 element.

### **Lasso**

#### Parameters:

- `alpha`: {float, ndarray of shape `(n_targets,)`}, default=1.0, if an array is passed, penalties are assumed to be specific to the targets. Hence they must correspond in number.
- `fit_intercept`: default=True (if set to false then data is expected to be centred)
- `normalize`: default=False, ignored if `fit_intercept` is set to False, regressors will be normalized before regression by subtracting the mean and dividing by the l2-norm.

- precompute: array-like of shape (n\_features, n\_features), default=False
- copy\_X: default=True, X is copied otherwise overwritten.
- max\_iter: int, default=1000 number of iterations for gradient descent.
- tol: precision of the solution default=1e-4
- warm\_start: reuse the solution of the previous call to fit as initialization
- positive: forces coefficients to be positive
- random\_state: int, RandomState instance, default=None
- selection: default: cyclic describes how the coefficient will be updated

Attributes:

- coef\_ : array of shape (n\_features, ) or (n\_targets, n\_features) returns the array containing all the theta values calculated for the linear regression
- dual\_gap\_: given alpha dual gaps at the end of optimization
- Sparse\_coef\_: sparse representation of the fitted coefficients.
- intercept\_: float or array of shape (n\_targets,) returns independent term in the linear model. Set to 0.0 if fit\_intercept = False.
- n\_iter: ndarray of shape (n\_classes,) or (1, ), actual number of iterations for all classes. If binary or multinomial, it returns only 1 element.