Experiment 4 – Scikit Learn API Write-up

Name: Ashish Nanda

Roll No.: J041

1. Linear Regression

LinearRegression 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.

sklearn.linear_model.LinearRegression(*, fit_intercept=True, normalize=False, copy_X=True, n_jobs=None, positive=False)

Parameters

1. fit_intercept : bool, default=True
2. normalize : bool, default=False
3. copy_X : bool, default=True
4. n_jobs : int, default=None
5. positive : bool, default=False

Attributes

1. coef_array of shape (n_features,) or (n_targets, n_features)
2. rank_int
3. singular_array of shape (min(X, y),)
4. intercept_float or array of shape (n_targets,)

2. Logistic Regression

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)

Parameters

1. Penalty : {‘l1’, ‘l2’, ‘elasticnet’, ‘none’}, default=’l2’
2. dual : bool, default=False
3. tol : float, default=1e-4
4. C : float, default=1.0
5. fit_intercept : bool, default=True
6. intercept_scaling : float, default=1
7. class_weightdict or ‘balanced’, default=None
8. random_state : int, RandomState instance, default=None
9. solver{‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’}, default=’lbfgs’
10. max_iter : int, default=100
11. multi_class{‘auto’, ‘ovr’, ‘multinomial’}, default=’auto’
12. verbose : int, default=0
13. warm_start : bool, default=False
14. n_jobs : int, default=None
15. l1_ratio : float, default=None

Attributes

1. classes_ndarray of shape (n_classes,)
2. coef_ndarray of shape (1, n_features) or (n_classes, n_features)
3. intercept_ndarray of shape (1,) or (n_classes,)
4. n_iter_ndarray of shape (n_classes,) or (1,)

3. Ridge Regression

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)

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)).

Parameters

1. alpha{float, ndarray of shape (n_targets,)}, default=1.0
2. fit_intercept: bool, default=True
3. normalize: bool, default=False
4. copy_X: bool, default=True
5. max_iter: int, default=None
6. tol: float, default=1e-3
7. solver{‘auto’, ‘svd’, ‘cholesky’, ‘lsqr’, ‘sparse_cg’, ‘sag’, ‘saga’}, default=’auto’
8. random_state: int, RandomState instance, default=None

Attributes

1. coef_ndarray of shape (n_features,) or (n_targets, n_features)
2. intercept_float or ndarray of shape (n_targets,)
3. n_iter_None or ndarray of shape (n_targets,)

4. Lasso Regression

sklearn.linear_model.Lasso(alpha=1.0, *, fit_intercept=True, nor malize=False, precompute=False, copy_X=True, max_iter=1000, tol= 0.0001, warm_start=False, positive=False, random_state=None, select ion='cyclic')

Parameters

1. alpha: float, default=1.0
2. fit_intercept: bool, default=True
3. normalize: bool, default=False
4. precompute: bool or array-like of shape (n_features, n_features), default=False
5. copy_X: bool, default=True
6. max_iter: int, default=1000
7. tol: float, default=1e-4
8. warm_start: bool, default=False
9. positive: bool, default=False

10. random_state : int, RandomState instance, default=None

11. selection{‘cyclic’, ‘random’}, default=’cyclic’

Attributes

1. coef_ndarray of shape (n_features,) or (n_targets, n_features)
2. dual_gap_float or ndarray of shape (n_targets,)
3. sparse_coef_sparse matrix of shape (n_features, 1) or (n_targets, n_features)
4. intercept_float or ndarray of shape (n_targets,)
5. n_iter_int or list of int