SVR

class sklearn.svm.SVR(*, kernel='rbf', degree=3, gamma='scale', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache_size=200, verbose=False, max_iter=- 1)

Parameters

- kernel: {‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’}, default=’rbf’
- degree: int, default=3
- gamma: {‘scale’, ‘auto’} or float, default=’scale’
- coef0: float, default=0.0
- tol: float, default=1e-3
- C: float, default=1.0
- epsilon: float, default=0.1
- shrinking: bool, default=True
- cache_size: float, default=200
- verbose: bool, default=False
- max_iter: int, default=-1

Attributes

- class_weight_ndarray of shape (n_classes,)
- coef_ndarray of shape (1, n_features)
- dual_coef_ndarray of shape (1, n_SV)
- fit_status_int
- intercept_ndarray of shape (1,)
- n_support_ndarray of shape (n_classes,), dtype=int32
- shape_fit_tuple of int of shape (n_dimensions_of_X,)
- support_ndarray of shape (n_SV,)
- support_vectors_ndarray of shape (n_SV, n_features)

Working of SVR

- Support Vector Regression is a supervised learning algorithm that is used to predict discrete values.
- The basic idea behind SVR is to find the best fit line, it is the hyperplane that has the maximum number of points.
- Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value.
- The threshold value is the distance between the hyperplane and boundary line.
- The fit time complexity of SVR is more than quadratic with the number of samples which makes it hard to scale to datasets with more than a couple of 10000 samples.
- For large datasets, Linear SVR or SGD Regressor is used. Linear SVR provides a faster implementation than SVR but only considers the linear kernel.
- The model produced by Support Vector Regression depends only on a subset of the training data, because the cost function ignores samples whose prediction is close to their target.