

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)
```

kernel - Kernel to be used

degree - Degree of polynomial in case of polynomial kernel

gamma - Kernel coefficient in case of polynomial, rbf or sigmoid kernel

coef0 – Independent term in kernel function

C – Regularization parameter

epsilon- It specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value.

shrinking- Whether to use shrinking heuristic

max_iter- Limit on number of iterations to use

Attributes

class_weight - Multipliers of parameter C for each class. Computed based on the `class_weight` parameter.

coef - Weights assigned to the features (coefficients in the primal problem). This is only available in the case of a linear kernel.

dual_coef - Coefficients of the support vector in the decision function.

fit_status - 0 if correctly fitted, 1 otherwise (will raise warning)

intercept - constants

n_support - Number of support vectors for each class.

support_vectors - Support vectors

Methods

<code>fit(X, y[, sample_weight])</code>	Fit the SVM model according to the given training data.
<code>get_params([deep])</code>	Get parameters for this estimator.
<code>predict(X)</code>	Perform regression on samples in X.
<code>score(X, y[, sample_weight])</code>	Return the coefficient of determination R2 of the prediction.
<code>set_params(**params)</code>	Set the parameters of this estimator.

How SVR works?

- i. Support Vector Regression is a supervised learning algorithm that is used to predict discrete values.
- ii. The basic idea behind SVR is to find the best fit line, it is the hyperplane that has the maximum number of points.
- iii. 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.
- iv. The threshold value is the distance between the hyperplane and boundary line.
- v. 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.
- vi. For large datasets, Linear SVR or SGD Regressor is used. Linear SVR provides a faster implementation than SVR but only considers the linear kernel.
- vii. 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.