Support Vector Machines

**Support vector machines (SVMs)** are a set of supervised learning methods used for [classification](https://scikit-learn.org/stable/modules/svm.html#svm-classification), [regression](https://scikit-learn.org/stable/modules/svm.html#svm-regression) and [outliers detection](https://scikit-learn.org/stable/modules/svm.html#svm-outlier-detection).

The advantages of support vector machines are:

* Effective in high dimensional spaces.
* Still effective in cases where number of dimensions is greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

* If the number of features is much greater than the number of samples, avoid over-fitting in choosing [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) and regularization term is crucial.
* SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see [Scores and probabilities](https://scikit-learn.org/stable/modules/svm.html#scores-probabilities), below).

The support vector machines in scikit-learn support both dense (numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. However, to use an SVM to make predictions for sparse data, it must have been fit on such data. For optimal performance, use C-ordered numpy.ndarray (dense) or scipy.sparse.csr\_matrix (sparse) with dtype=float64.

The method of Support Vector Classification can be extended to solve regression problems. This method is called Support Vector Regression.

The model produced by support vector classification (as described above) depends only on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, 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.

There are three different implementations of Support Vector Regression: [**SVR**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR), **[NuSVR](https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVR.html" \l "sklearn.svm.NuSVR" \o "sklearn.svm.NuSVR)** and **[LinearSVR](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html" \l "sklearn.svm.LinearSVR" \o "sklearn.svm.LinearSVR)**. **[LinearSVR](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html" \l "sklearn.svm.LinearSVR" \o "sklearn.svm.LinearSVR)** provides a faster implementation than [**SVR**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR) but only considers the linear kernel, while **[NuSVR](https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVR.html" \l "sklearn.svm.NuSVR" \o "sklearn.svm.NuSVR)** implements a slightly different formulation than [**SVR**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR) and **[LinearSVR](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html" \l "sklearn.svm.LinearSVR" \o "sklearn.svm.LinearSVR)**. See [Implementation details](https://scikit-learn.org/stable/modules/svm.html#svm-implementation-details) for further details.

As with classification classes, the fit method will take as argument vectors X, y, only that in this case y is expected to have floating point values instead of integer values:

>>>

**>>> from** **sklearn** **import** svm

**>>>** X = [[0, 0], [2, 2]]

**>>>** y = [0.5, 2.5]

**>>>** regr = svm.SVR()

**>>>** regr.fit(X, y)

SVR()

**>>>** regr.predict([[1, 1]])

array([1.5])

This is just a very, very brief overview of these concepts and for more information on it I encourage you to check out the following webpage https://www.probabilisticworld.com/frequentist-bayesian-approaches-inferential-statistics/