SVM write up:

Code: *class* sklearn.svm.SVC(*\**, *C=1.0*, *kernel='rbf'*, *degree=3*, *gamma='scale'*, *coef0=0.0*, *shrinking=True*, *probability=False*, *tol=0.001*, *cache\_size=200*, *class\_weight=None*, *verbose=False*, *max\_iter=- 1*, *decision\_function\_shape='ovr'*, *break\_ties=False*, *random\_state=None*)

-Support Vector Classification.

The implementation is based on libsvm. The fit time scales at least quadratically with the number of samples and may be impractical beyond tens of thousands of samples. For large datasets consider using [**LinearSVC**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC) or **[SGDClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html" \l "sklearn.linear_model.SGDClassifier" \o "sklearn.linear_model.SGDClassifier)** instead, possibly after a **[Nystroem](https://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.Nystroem.html" \l "sklearn.kernel_approximation.Nystroem" \o "sklearn.kernel_approximation.Nystroem)** transformer.

The multiclass support is handled according to a one-vs-one scheme.

For details on the precise mathematical formulation of the provided kernel functions and how gamma, coef0 and degree affect each other, see the corresponding section in the narrative documentation: Kernel functions.

**C*float, default=1.0***

Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty.

**kernel*{‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’}, default=’rbf’***

Specifies the kernel type to be used in the algorithm. It must be one of ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’ or a callable. If none is given, ‘rbf’ will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that matrix should be an array of shape (n\_samples, n\_samples).

**degree*int, default=3***

Degree of the polynomial kernel function (‘poly’). Ignored by all other kernels.

**gamma*{‘scale’, ‘auto’} or float, default=’scale’***

Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’.

* if gamma='scale' (default) is passed then it uses 1 / (n\_features \* X.var()) as value of gamma,
* if ‘auto’, uses 1 / n\_features.

*Changed in version 0.22:*The default value of gamma changed from ‘auto’ to ‘scale’.

**coef0*float, default=0.0***

Independent term in kernel function. It is only significant in ‘poly’ and ‘sigmoid’.

**shrinking*bool, default=True***

Whether to use the shrinking heuristic. See the [User Guide](https://scikit-learn.org/stable/modules/svm.html#shrinking-svm).

**probability*bool, default=False***

Whether to enable probability estimates. This must be enabled prior to calling fit, will slow down that method as it internally uses 5-fold cross-validation, and predict\_proba may be inconsistent with predict. Read more in the [User Guide](https://scikit-learn.org/stable/modules/svm.html#scores-probabilities).

**tol*float, default=1e-3***

Tolerance for stopping criterion.

**cache\_size*float, default=200***

Specify the size of the kernel cache (in MB).

**class\_weight*dict or ‘balanced’, default=None***

Set the parameter C of class i to class\_weight[i]\*C for SVC. If not given, all classes are supposed to have weight one. The “balanced” mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n\_samples / (n\_classes \* np.bincount(y)).

**verbose*bool, default=False***

Enable verbose output. Note that this setting takes advantage of a per-process runtime setting in libsvm that, if enabled, may not work properly in a multithreaded context.

**max\_iter*int, default=-1***

Hard limit on iterations within solver, or -1 for no limit.

**decision\_function\_shape*{‘ovo’, ‘ovr’}, default=’ovr’***

Whether to return a one-vs-rest (‘ovr’) decision function of shape (n\_samples, n\_classes) as all other classifiers, or the original one-vs-one (‘ovo’) decision function of libsvm which has shape (n\_samples, n\_classes \* (n\_classes - 1) / 2). However, one-vs-one (‘ovo’) is always used as multi-class strategy. The parameter is ignored for binary classification.

*Changed in version 0.19:*decision\_function\_shape is ‘ovr’ by default.

*New in version 0.17: decision\_function\_shape=’ovr’* is recommended.

*Changed in version 0.17:*Deprecated *decision\_function\_shape=’ovo’ and None*.

**break\_ties*bool, default=False***

If true, decision\_function\_shape='ovr', and number of classes > 2, [predict](https://scikit-learn.org/stable/glossary.html#term-predict) will break ties according to the confidence values of [decision\_function](https://scikit-learn.org/stable/glossary.html" \l "term-decision_function); otherwise the first class among the tied classes is returned. Please note that breaking ties comes at a relatively high computational cost compared to a simple predict.

*New in version 0.22.*

**random\_state*int, RandomState instance or None, default=None***

Controls the pseudo random number generation for shuffling the data for probability estimates. Ignored when probability is False. Pass an int for reproducible output across multiple function calls. See [Glossary](https://scikit-learn.org/stable/glossary.html#term-random_state).

**Attributes**

**class\_weight\_*ndarray of shape (n\_classes,)***

Multipliers of parameter C for each class. Computed based on the class\_weight parameter.

**classes\_*ndarray of shape (n\_classes,)***

The classes labels.

[**coef\_**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.coef_)***ndarray of shape (n\_classes \* (n\_classes - 1) / 2, n\_features)***

Weights assigned to the features when kernel="linear".

**dual\_coef\_*ndarray of shape (n\_classes -1, n\_SV)***

Dual coefficients of the support vector in the decision function (see [Mathematical formulation](https://scikit-learn.org/stable/modules/sgd.html#sgd-mathematical-formulation)), multiplied by their targets. For multiclass, coefficient for all 1-vs-1 classifiers. The layout of the coefficients in the multiclass case is somewhat non-trivial. See the [multi-class section of the User Guide](https://scikit-learn.org/stable/modules/svm.html#svm-multi-class) for details.

**fit\_status\_*int***

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

**intercept\_*ndarray of shape (n\_classes \* (n\_classes - 1) / 2,)***

Constants in decision function.

**n\_features\_in\_*int***

Number of features seen during [fit](https://scikit-learn.org/stable/glossary.html#term-fit).

*New in version 0.24.*

**feature\_names\_in\_*ndarray of shape (n\_features\_in\_,)***

Names of features seen during [fit](https://scikit-learn.org/stable/glossary.html#term-fit). Defined only when X has feature names that are all strings.

*New in version 1.0.*

**support\_*ndarray of shape (n\_SV)***

Indices of support vectors.

**support\_vectors\_*ndarray of shape (n\_SV, n\_features)***

Support vectors.

[**n\_support\_**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.n_support_)***ndarray of shape (n\_classes,), dtype=int32***

Number of support vectors for each class.

[**probA\_**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.probA_)***ndarray of shape (n\_classes \* (n\_classes - 1) / 2)***

Parameter learned in Platt scaling when probability=True.

[**probB\_**](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.probB_)***ndarray of shape (n\_classes \* (n\_classes - 1) / 2)***

Parameter learned in Platt scaling when probability=True.

**shape\_fit\_*tuple of int of shape (n\_dimensions\_of\_X,)***

Array dimensions of training vector X.