1.

**C : Regularization parameter** :-

The C parameter trades of correct classification of training examples against maximization of the decision functions margin. It is a control on fitting parameter. The magnitudes of the fitting parameter increases , there will be an increase in the penality of the cost function. For a larger values of C a smaller margin will be accepted , the decision function is better at classifying all training points correctly. The strength of the regularization is inversely proportional to C.

**Kernel** :-

Gives the idea about type of kernel used in the algorithm. Kernals help to determine the shape of the hyperplane and decision boundary. It is a way to solve non linear problems with the help of linear classifiers. kernels provide shortcuts to avoid complex calculations. The value of kernel can be any type from linear to polynomial. If the value of the kernel is linear then the decision boundary would be linear and two dimensional. RBF is the default kernel.

**Gamma** :-

Gamma is a parameter for non linear hyperplanes. The behavior of the model is very sensitive to the gamma parameter. If gamma is too large, the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting. When gamma is very small, the model is too constrained and cannot capture the complexity or “shape” of the data.

**Degree** :-

It is a parameter used when kernel is set to 'poly'. It is basically the degrees of the polynomial used to find the hyperplane to split the data.

**Coef0 :-**

Independent term in kernel function. It is only significant in polynomial and sigmoid kernals.

**Shrinking** :- This has to do with whether or not we want a shrinking heuristic used in the optimization of the svm. If the number of iterations is large, then shrinking can shorten the training time.

**Probability** :-

This parameter enables or disables probability estimates. The default value is false, but it must be enabled before call fit.

**tol :-**

Tolerance of stopping criteria. This tells scikit to stop searching for a minimum (or maximum) once some tolerance is achieved. tol will change depending on the objective function being minimized and the algorithm they use to find the minimum, and thus will depend on the model we are fitting.

**Cache\_size** :-

Specify the size of the kernel cache (in MB). The size of the kernel cache has a strong impact on run times for larger problems.

**class\_weight :-**

The class\_weight is a dictionary that defines each class label (e.g. 0 and 1) and the weighting to apply to the C value in the calculation of the soft margin.

**Verbose :-**

**By default Verbose is False .Enable verbose output. It takes the advantage of a per-process runtime setting in libsvm that, if enabled, may not work properly in a multithreaded context.**

**max\_iter :-**

**Max\_iter=-1 by default. This causes the optimizer to have no maximum number of iterations, and can cause the classifier to run very long,when solving hard problems.**

**decision\_function\_shape :-**

**Decision\_function\_shape is used** for multi- class classification. If n\_class is the number of classes, then n\_class \* (n\_class - 1) / 2 classifiers are constructed and each one trains data from two classes. To provide a consistent interface with other classifiers, the decision\_function\_shape option allows to aggregate the results of the “one-against-one” classifiers to a decision function of shape (n\_samples, n\_classes)

**break\_ties :-**

It is optional*,* By default it is false.

True − The predict will break ties according to the confidence values of decision\_function

False − The predict will return the first class among the tied classes.

Tie breaking is costly if decision\_function\_shape=’ovr’ and therefore it is not enabled by default.

***random\_state***:-

Controls the pseudo random number generation for shuffling the data for probability estimates.

This parameter represents the seed of the pseudo random number generated which is used while shuffling the data. Followings are the options −

* int − In this case, *random\_state* is the seed used by random number generator.
* RandomState instance − In this case, random\_state is the random number generator.

2.

Kernel function is a method used to take data as input and transform it into the required form of processing data. Kernel function generally transforms the training set data so that a non linear decision surface is able to transformed to a linear equation in to a higher dimensional space. The major kernel function used in SVM are Linear, Polynomial, RBF and sigmoid kernel.

**Linear Kernel** :-

Linear SVM is used for linear separable data, that is dataset can be easily separated by a straight line. Linear svm is commonly used in a dataset contains large number of features.

Equation :  z=y\*x+b

Advantages :-

* Memory efficient
* Faster than other kernels

Disadvantages :-

* Highest possibility of underfitting

Application :

Text classification

**Polynomial Kernel :-**

When the data in not linearly separable, make it linearly separable y mapping in to higher dimensional space. Well suited for problems all the training datas are normalized.

polynomial kernel svm is mainly used in image processing.

  Equation :- z=(y\*x+b)2

**RBF Kernel :-**

When the data is not linearly separable, make it linearly separable by mapping the data in to infinite dimensional space. When we have no previous information about the data we can use RBF kernel.

z=exp(-γ||χ-y ||2)

* Best learning time
* Best for fitting data
* Highest possibility of overfitting

**SIGMOID** **Kernel :**- Also known as Hyperbolic tangent kernel and multi layer perceptron kernel. The sigmoid kernel comes from the neural network filed where the bipolar sigmoid function is used as the activation function for artificial neurons. It has excellent classification accuracy. Sigmoid kernels has complicated structure.

Equation :- k(x, y) = tanh (alpha x^T y + c) 