Support Vector Machine

Support Vectors: These are the points that are closest to the hyperplane. A separating line will be defined with the help of these data points.

Margin: it is the distance between the hyperplane and the observations closest to the hyperplane (support vectors). In SVM large margin is considered a good margin. There are two types of margins **hard margin** and **soft margin.** I will talk more about these two in the later section.

So.. what is it?

It's a supervised machine learning problem where we try to find a hyperplane that best separates the two classes.

Similar to logistic regression, however logistic regression is based on probabilistic approaches while SVM is based on statistical approaches.

How do you choose the best hyperplane?

SVM does this by finding the maximum margin between the hyperplanes that means maximum distance between the two classes.

Whats a hyperplane?

In geometry, a hyperplane is a subspace whose dimension is one less than that of its ambient space. For example, if a space is 3-dimensional then its hyperplanes are the 2-dimensional planes, while if the space is 2-dimensional, its hyperplanes are the 1-dimensional lines.

When to use logistic regression vs Support vector machine?

It can be dependent on the number of features in your dataset. SVM works best when our dataset is small and complex. It does no harm in testing both. It can be

advisable to try logistic regression first to see how well it does. If the results are subpar you can try SVM without any kernel. Without a kernel both LR and SVM tend to have similar results, depending on your features one can be more / less efficient than the others.

Logistic regression and SVM without any kernel have similar performance but depending on your features, one may be more efficient than the other.

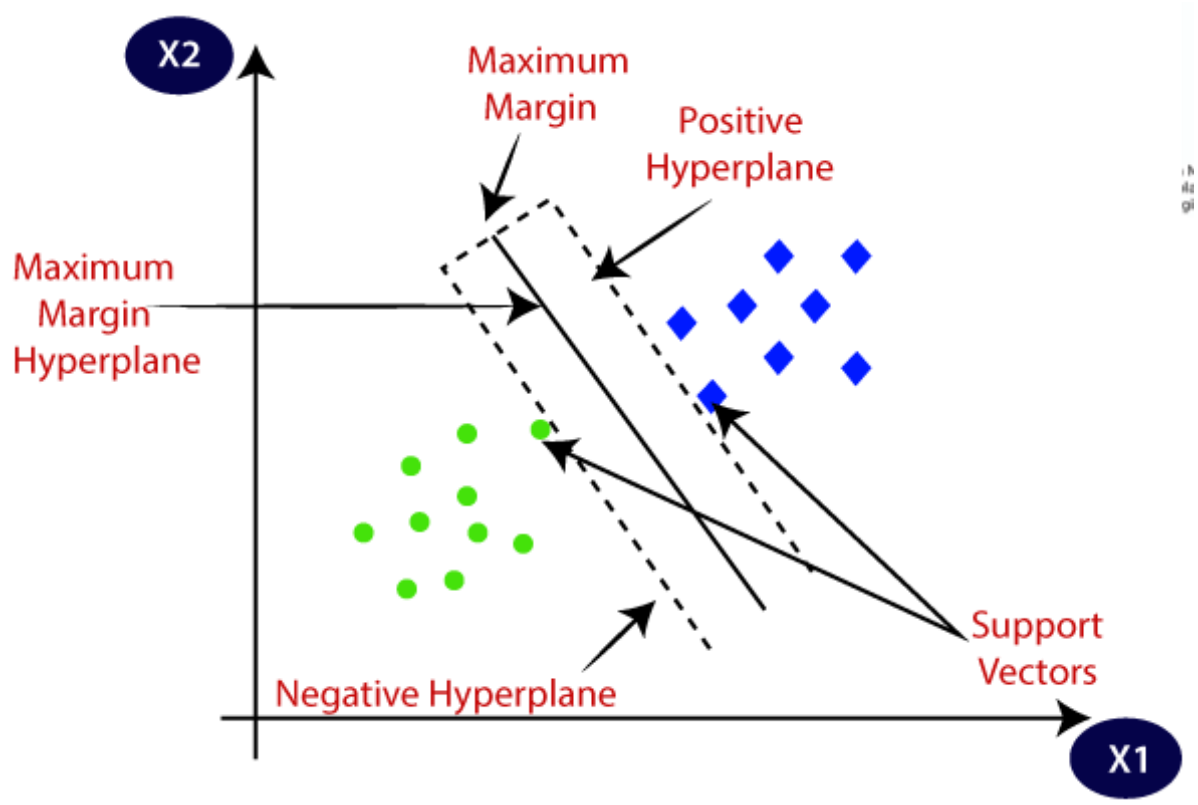
Types of Support Vector Machines.

Linear SVM

When our data is perfectly linearly separable we can use Linear SVM. This means the data can be separated into 2 classes using a single straight line (if its 2D)

Non-Linear SVM

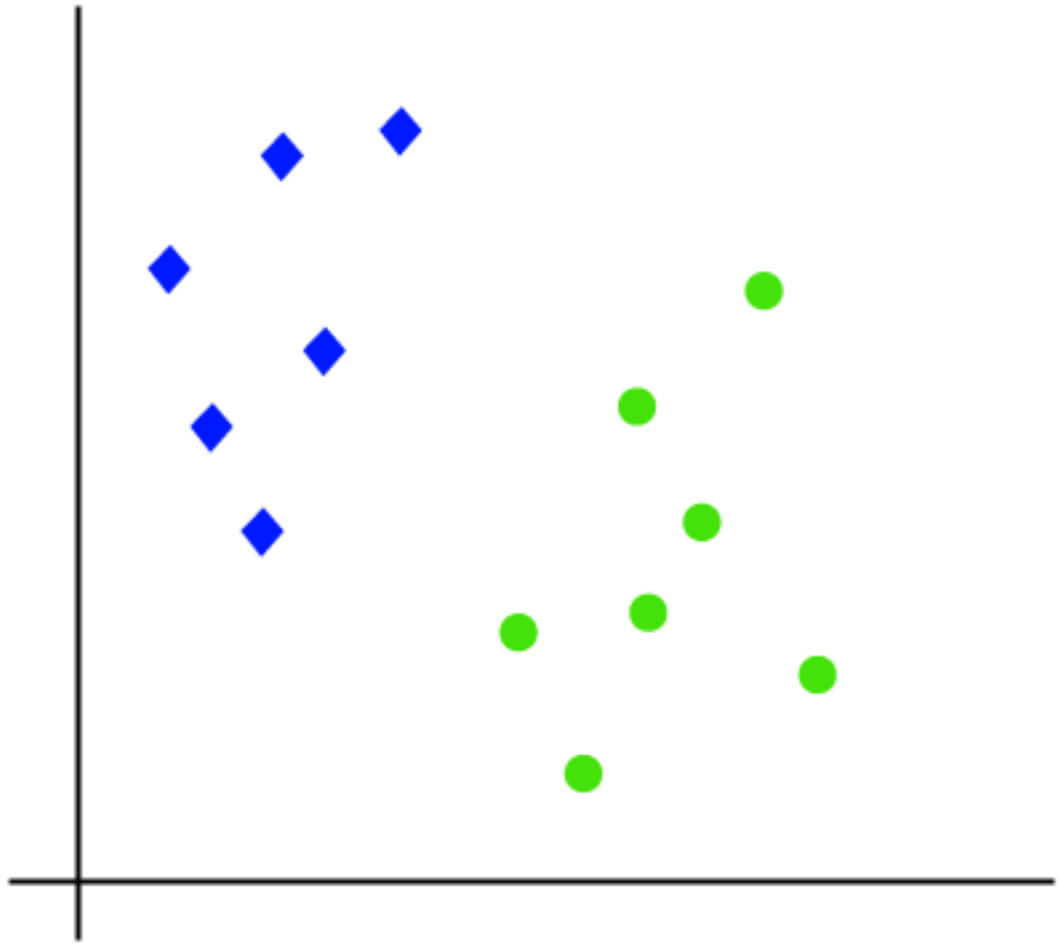
When the data is not linearly separable then we can use Non-Linear SVM, which means the data cannot be separated into 2 classes by using a straight line (if 2D), we can employ some techniques like kernels to classify them. Of course in real world applications we don't fin linearly separable classes.



How does SVM Work?

SVM is defined in terms of the support vectors only, we don't have to worry about other observations since the margin is made using the points which are closest to the hyper plane (support vectors). Compared t logistic regression which is defined in terms of all points it enjoys some speed advantages.

Let's understand the working of SVM using an example. Suppose we have a dataset that has two classes (green and blue). We want to classify that the new data point as either blue or green.



To classify these points, we can have many decision boundaries, but the question is which is the best and how do we find it? **NOTE:** Since we are plotting the data points in a 2-dimensional graph we call this decision boundary a **straight line** but if we have more dimensions, we call this decision boundary a **"hyperplane"**

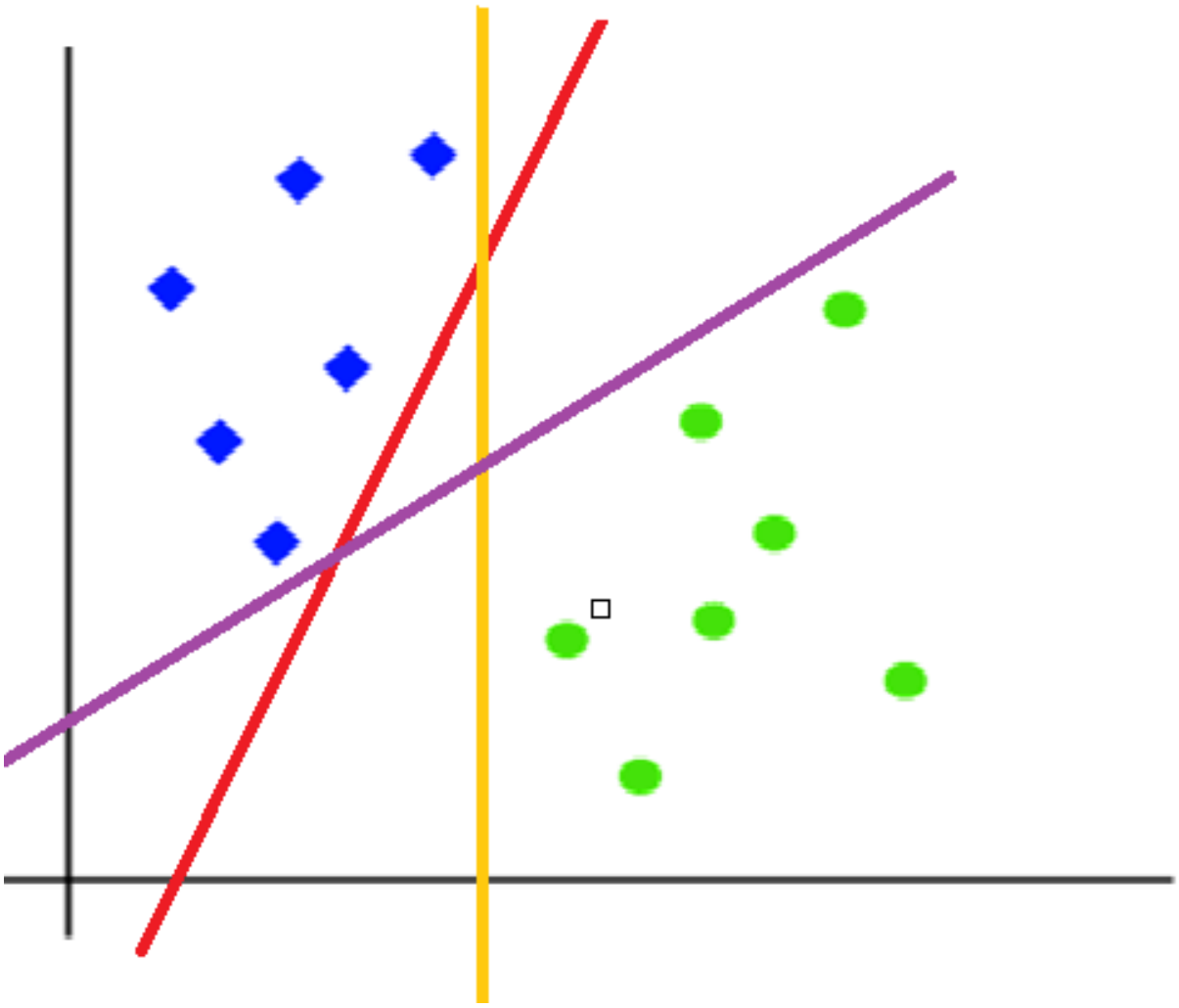
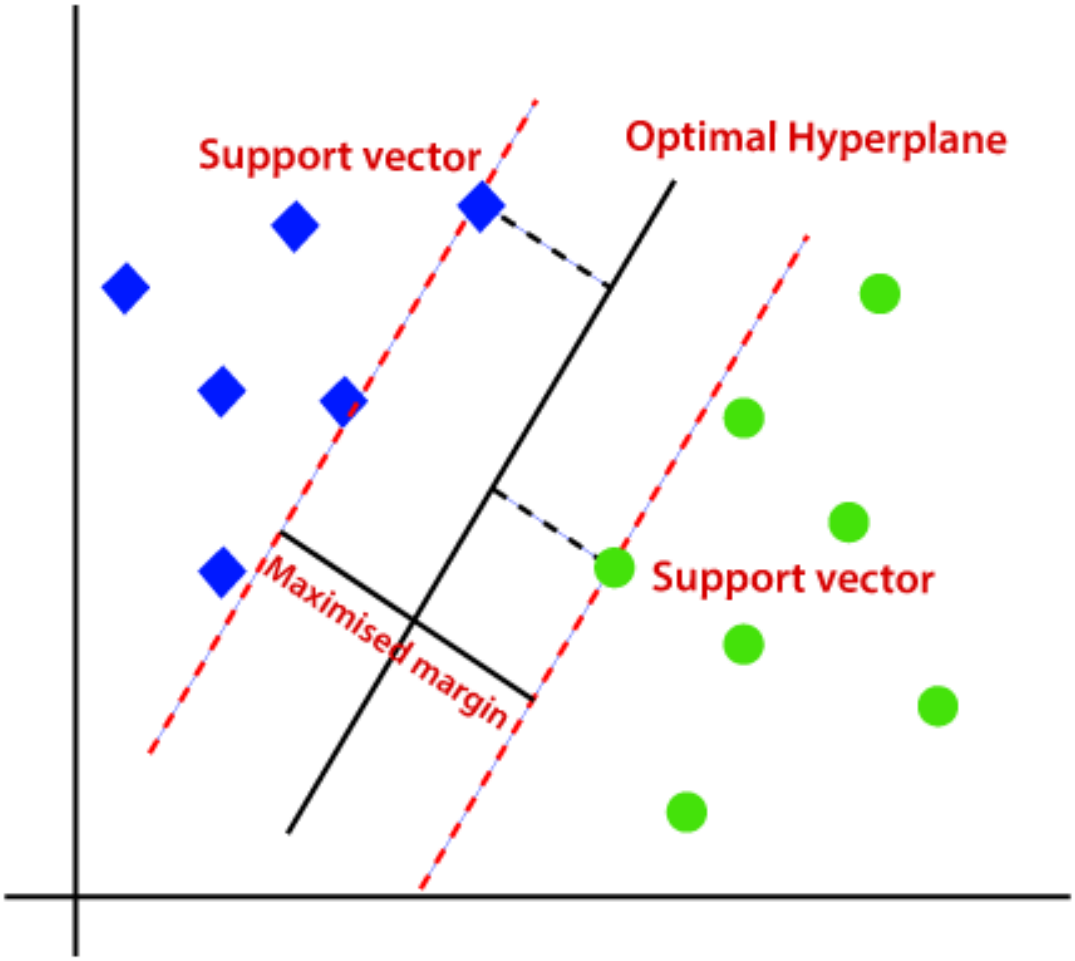


Image Source: Author

The best hyperplane is that plane that has the maximum distance from both the classes, and this is the main aim of SVM. This is done by finding different hyperplanes which classify the labels in the best way then it will choose the one which is farthest from the data points or the one which has a maximum margin.



The difference between a hard margin and a soft margin in SVMs lies in the separability of the data. **If our data is linearly separable, we go for a hard margin**. However, if this is not the case, it won't be feasible to do that.

Kernels in Support Vector Machine

The most interesting feature of SVM is that it can even work with a non-linear dataset and for this, we use “Kernel Trick” which makes it easier to classifies the points. Suppose we have a dataset like this:

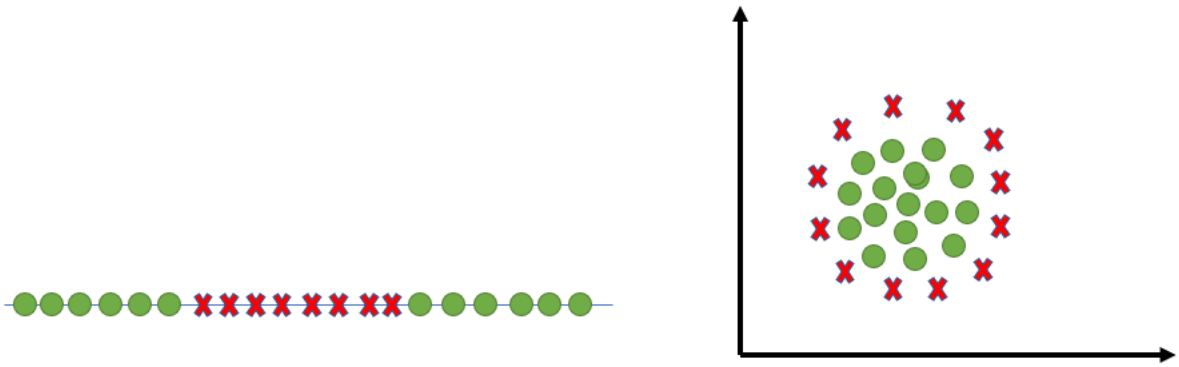


Image Source: Author

Here we see we cannot draw a single line or say hyperplane which can classify the points correctly. So what we do is try converting this lower dimension space to a higher dimension space using some quadratic functions which will allow us to find a decision boundary that clearly divides the data points. These functions which help us do this are called Kernels and which kernel to use is purely determined by hyperparameter tuning.

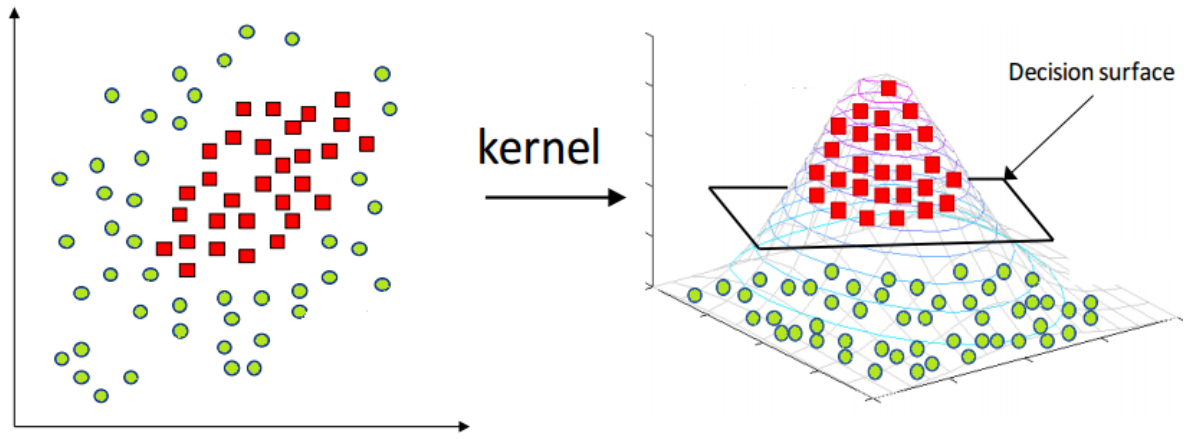


Image 3

Different Kernel functions

Some kernel functions which you can use in SVM are given below:

1. Polynomial kernel

Following is the formula for the polynomial kernel:

$$f(X1, X2) = (X1^T . X2 + 1)^d$$

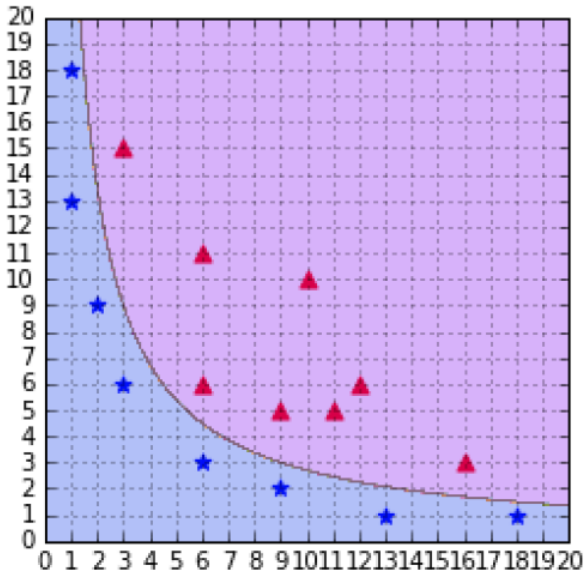
Here d is the degree of the polynomial, which we need to specify manually.

Suppose we have two features X1 and X2 and output variable as Y, so using polynomial kernel we can write it as:

$$X1^T . X2 = \begin{bmatrix} X1 \\ X2 \end{bmatrix} . [X1 \quad X2]$$

$$= \begin{bmatrix} X1^2 & X1.X2 \\ X1.X2 & X2^2 \end{bmatrix}$$

So we basically need to find X_1^2 , X_2^2 and X1.X2, and now we can see that 2 dimensions got converted into 5 dimensions.



A SVM using a polynomial kernel is able to separate the data (degree=2)

Image 4

2. Sigmoid kernel

We can use it as the proxy for neural networks. Equation is:

$$f(x1, x2) = \tanh(\alpha x^T y + x)$$

It is just taking your input, mapping them to a value of 0 and 1 so that they can be separated by a simple straight line.

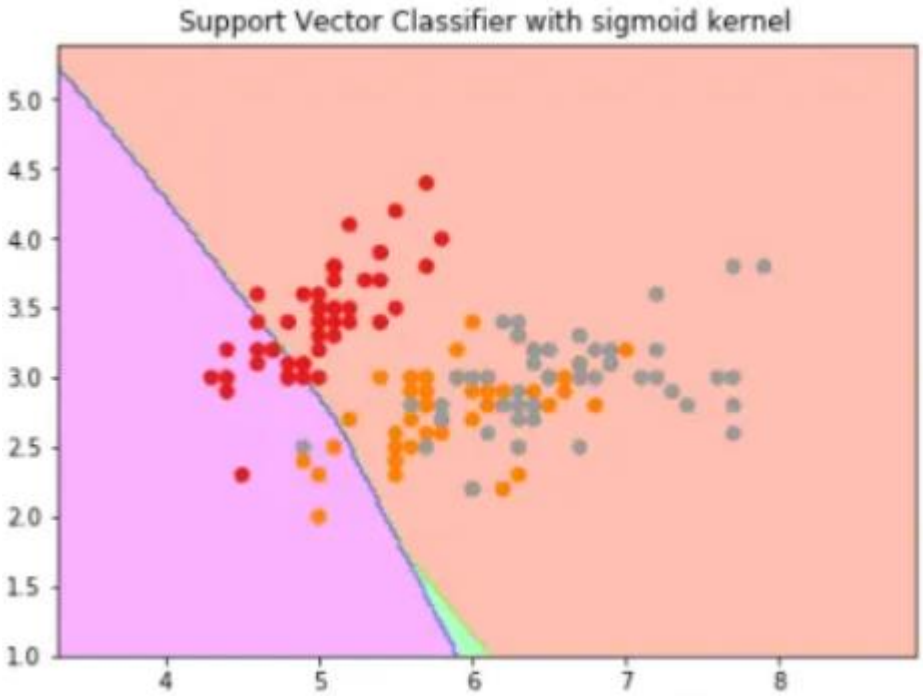


Image Source: https://dataaspirant.com/svm-kernels/#t-1608054630725

3. RBF kernel

What it actually does is to create non-linear combinations of our features to lift your samples onto a higher-dimensional feature space where we can use a linear decision boundary to separate your classes It is the most used kernel in SVM classifications, the following formula explains it mathematically:

$$f(x1, x2) = e^{\frac{-||(x1 - x2)||^2}{2\sigma^2}}$$

where,

1. 'σ' is the variance and our hyperparameter
2. $||X_1 - X_2||$ is the Euclidean Distance between two points X_1 and X_2

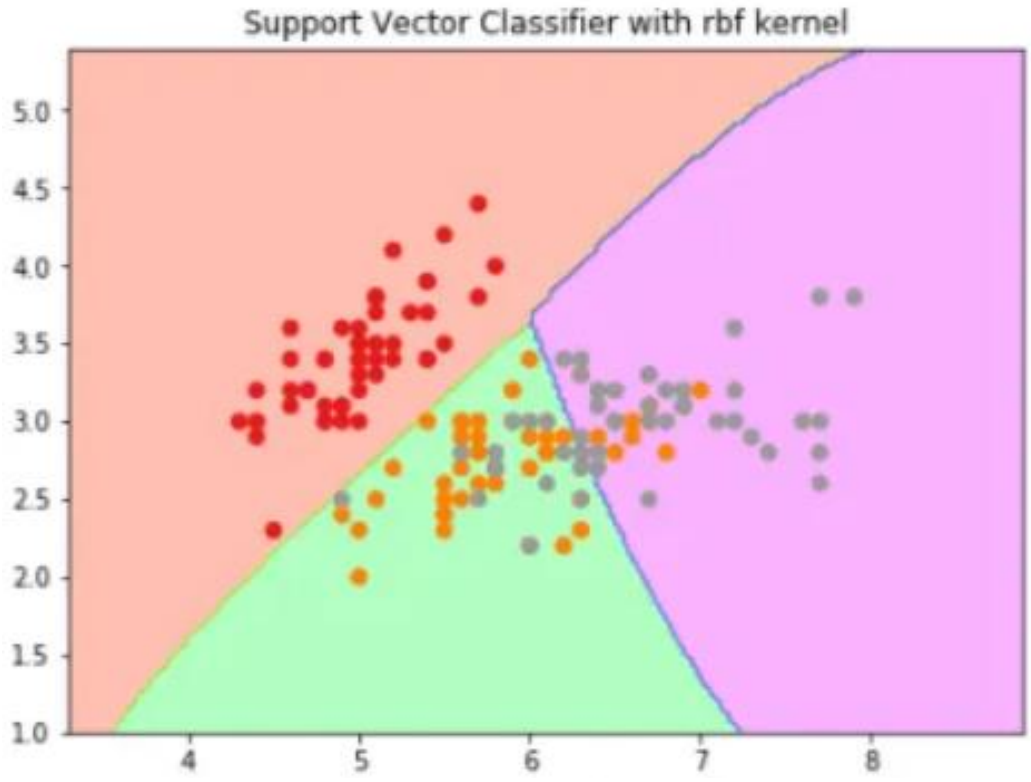


Image 5

4. Bessel function kernel

It is mainly used for eliminating the cross term in mathematical functions. Following is the formula of the Bessel function kernel:

$$k(x,y) = \frac{J_{v+1}(\sigma\|x-y\|)}{\|x-y\|^{-n(v+1)}}$$

5. Anova Kernel

It performs well on multidimensional regression problems. The formula for this kernel function is:

$$k(x,y) = \sum_{k=1}^{n} \exp(-\sigma(x^k - y^k)^2)^d$$

How to choose the right Kernel?

The kernel you choose is dataset dependent.
If it is linearly separable then you must opt. for linear kernel function since it is very easy to use and the complexity is much lower compared to other kernel functions. I'd recommend you start with a hypothesis that your data is linearly separable and choose a linear kernel function.

You can then work your way up towards the more complex kernel functions. Usually, we use SVM with RBF and linear kernel function because other kernels like polynomial kernel are rarely used due to poor efficiency. But what if linear and RBF both give approximately similar results? Which kernel do we choose now? Let's understand this with the help of an example, for simplicity I'll only take 2 features that mean 2 dimensions only. In the figure below I have plotted the decision boundary of a linear SVM on 2 features of the iris dataset:

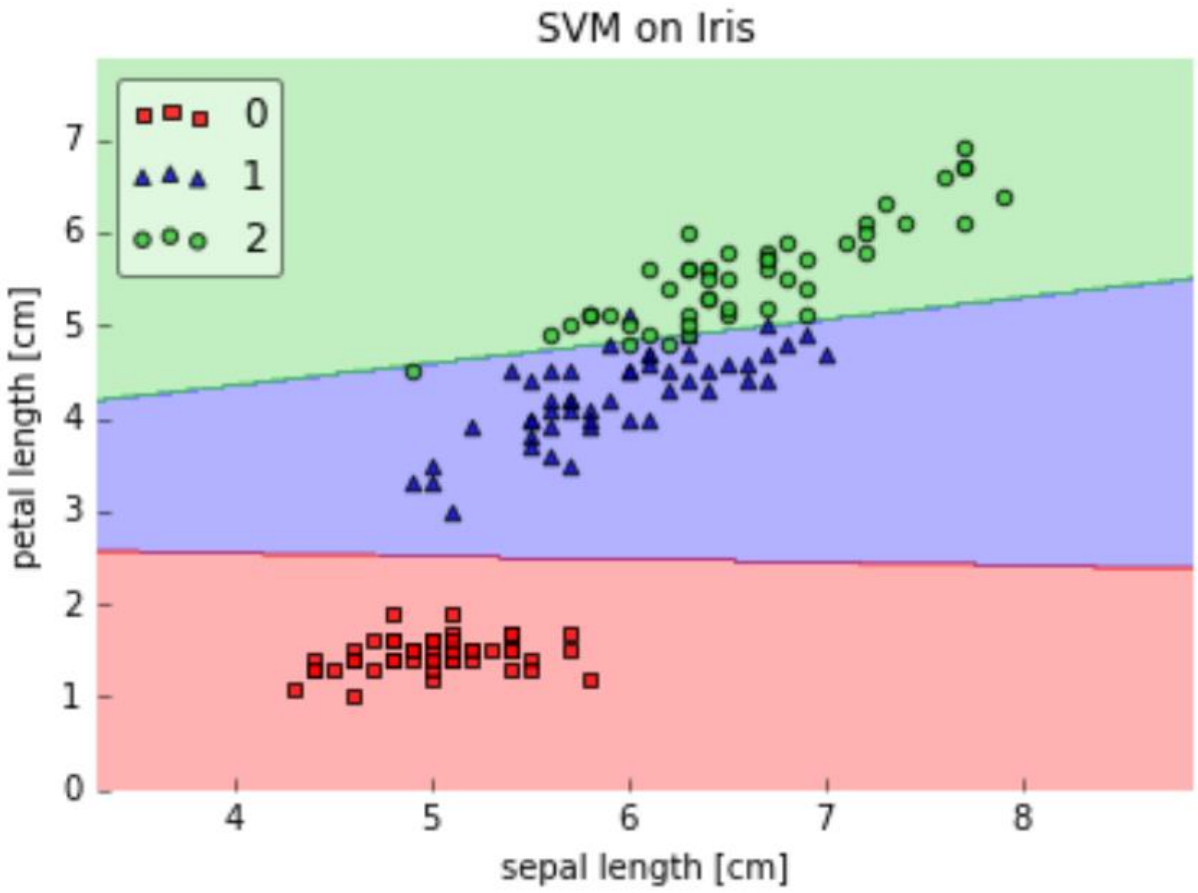


Image 6

Here we see that a linear kernel works fine on this dataset, but now let's see how will RBF kernel work.

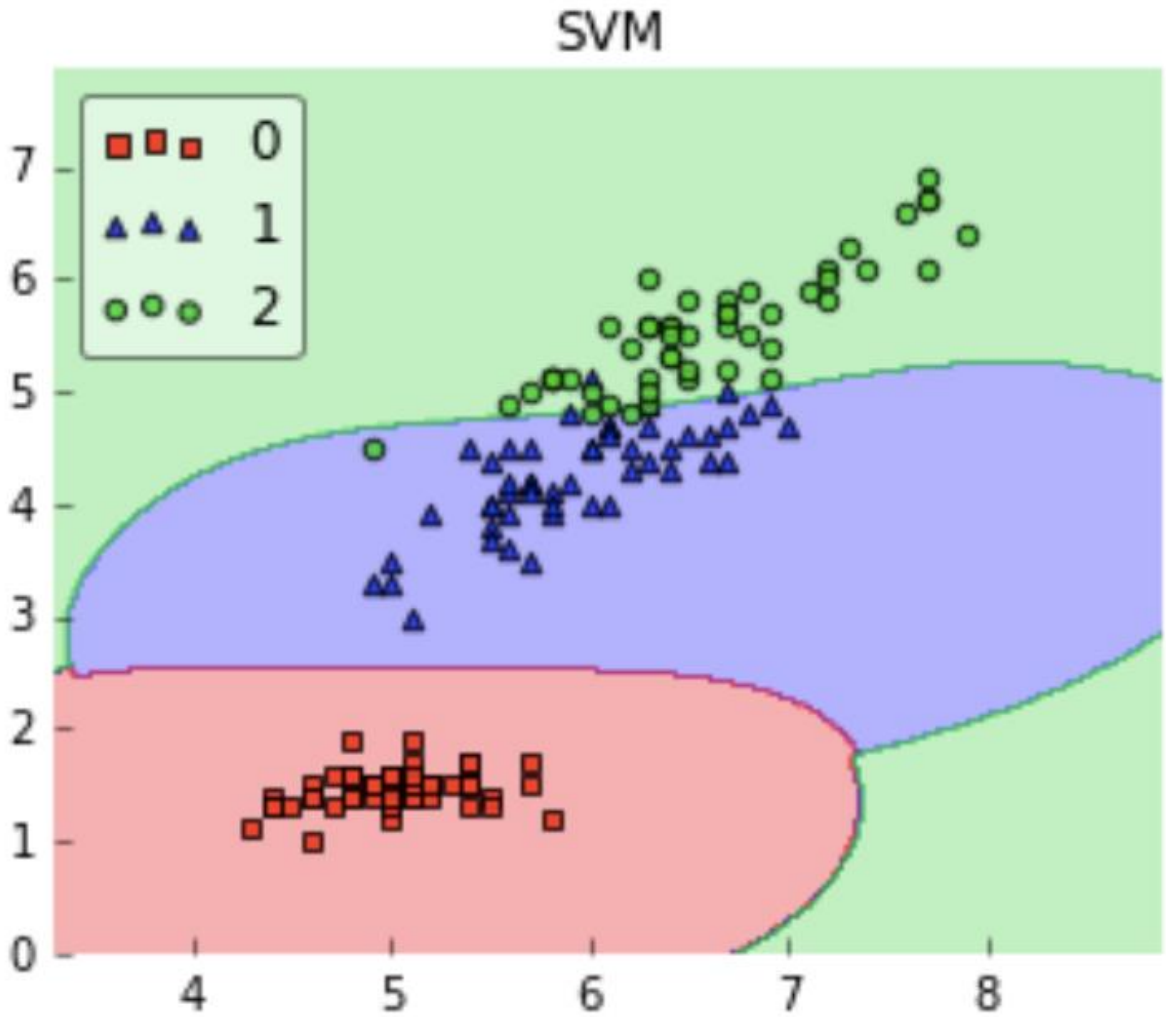


Image 7

We can observe that both the kernels give similar results, both work well with our dataset but which one should we choose? Linear SVM is a parametric model. A Parametric Model is a concept used to describe a model in which all its data is represented within its parameters. In short, the only information needed to predict the future from the current value is the parameters.

The complexity of the RBF kernel grows as the training data size increases. In addition to the fact that it is more expensive to prepare RBF kernel, we also have to keep the kernel matrix around, and the projection into this “infinite” higher dimensional space where the data becomes linearly separable is more expensive as well during prediction. If the dataset is not linear then using linear kernel doesn’t make sense we’ll get a very low accuracy if we do so.

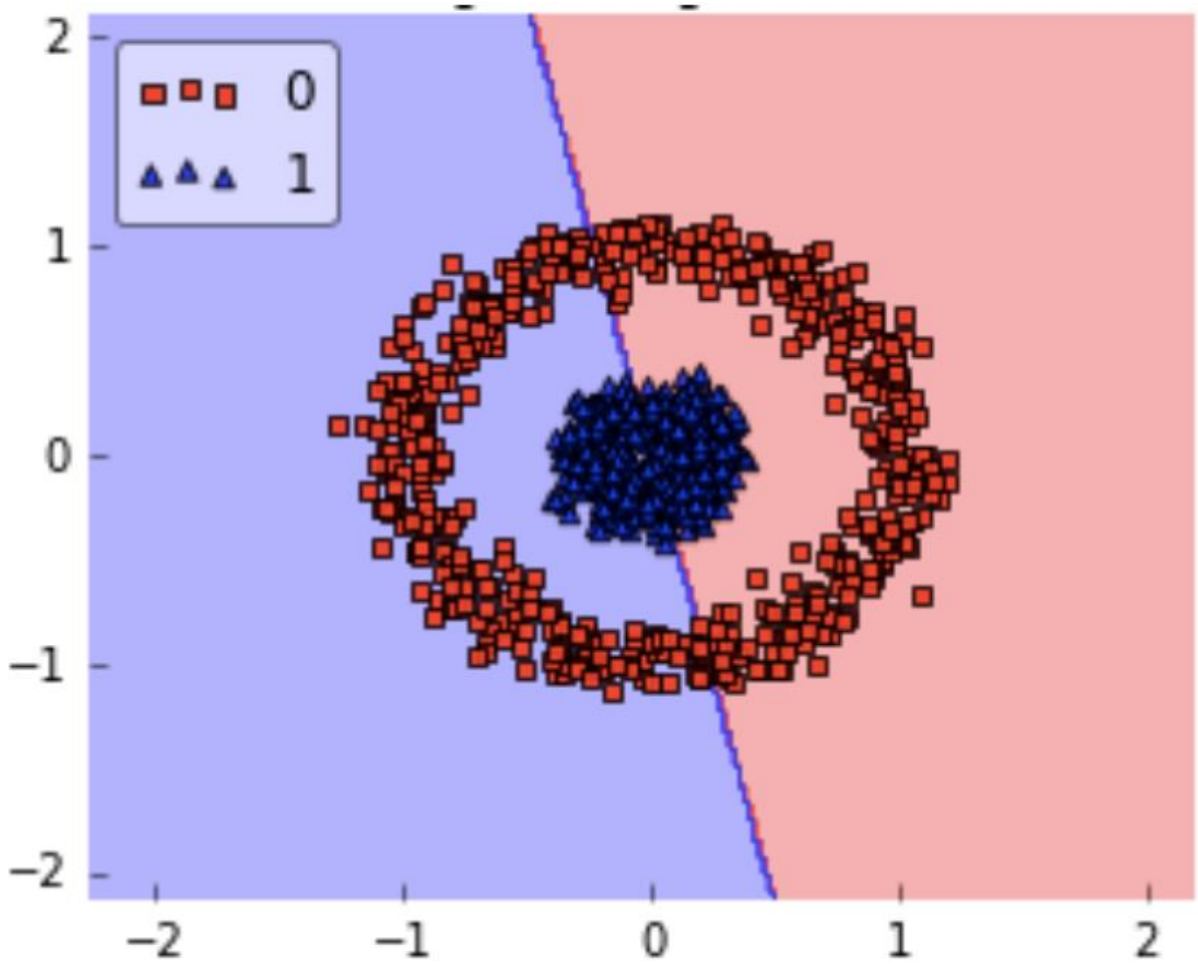


Image 8

So for this kind of dataset, we can use RBF without even a second thought because it makes decision boundary like this:

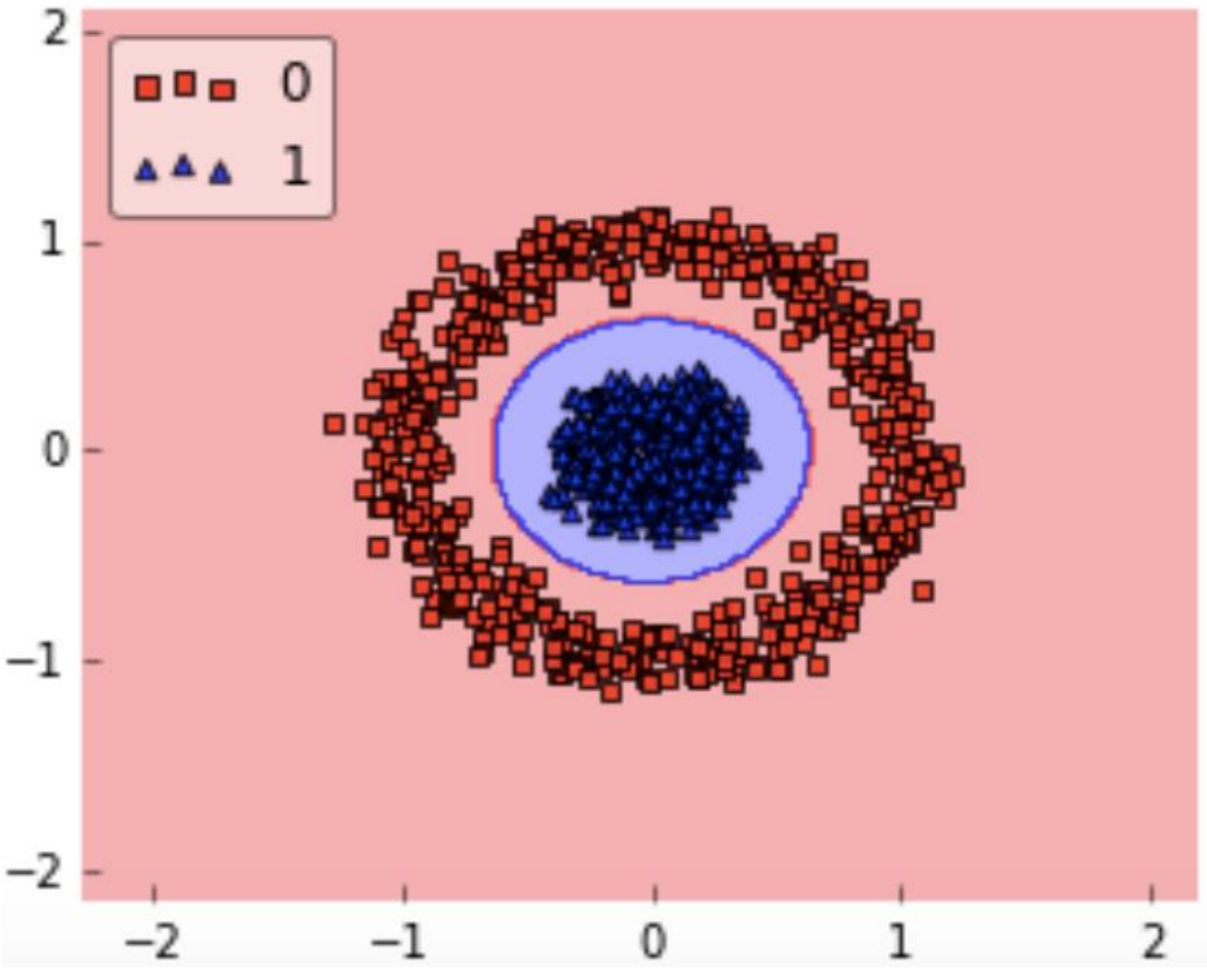


Image 9

Advantages of SVM

1. SVM works better when the data is Linear

2. It is more effective in high dimensions

3. With the help of the kernel trick, we can solve any complex problem

4. SVM is not sensitive to outliers

5. Can help us with Image classification

Disadvantages of SVM

1. Choosing a good kernel is not easy

2. It doesn't show good results on a big dataset

3. The SVM hyperparameters are Cost -C and gamma. It is not that easy to fine-tune these hyper-parameters. It is hard to visualize their impact