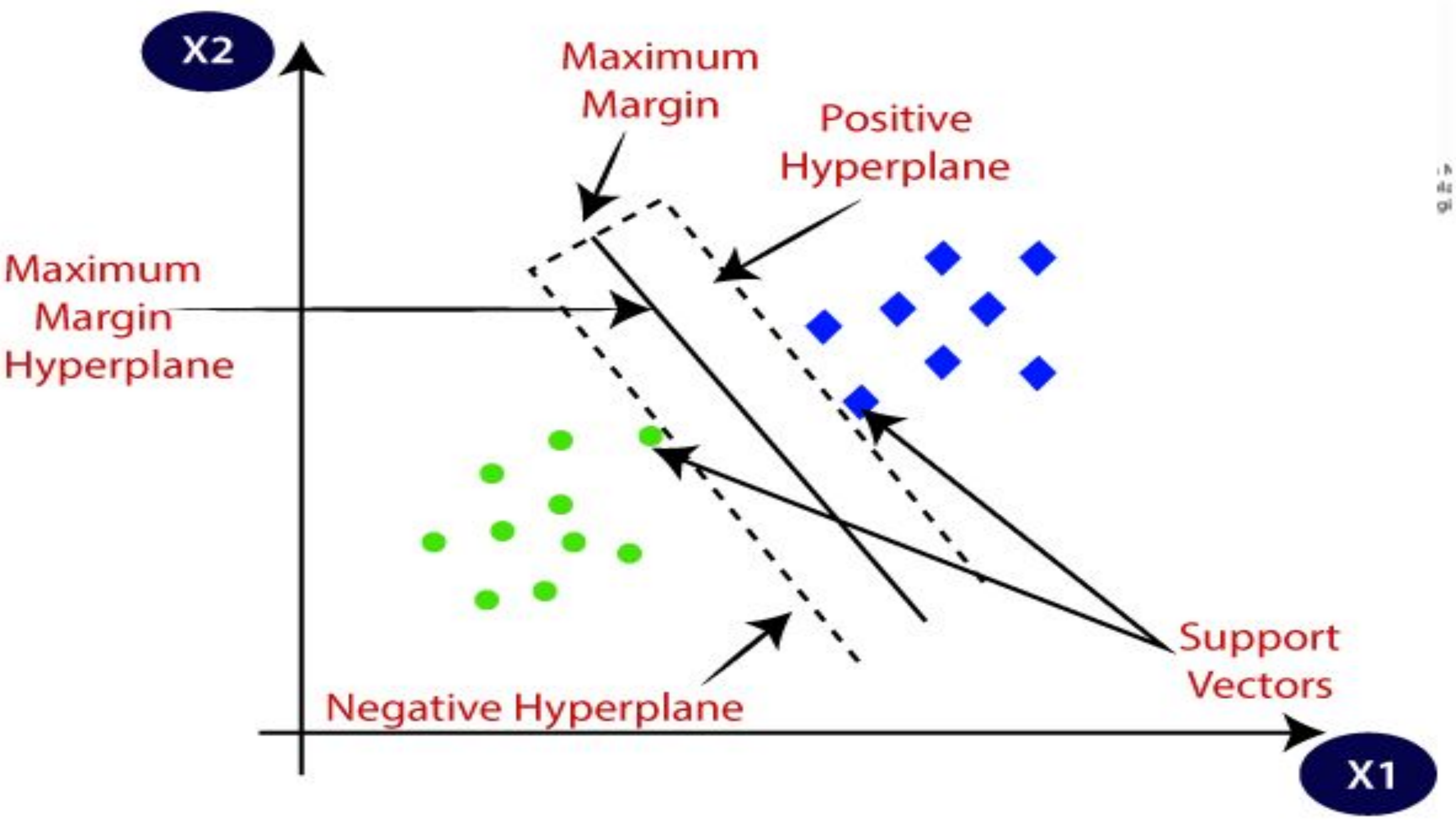
Support Vector Machines

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane



Support Vector Machines

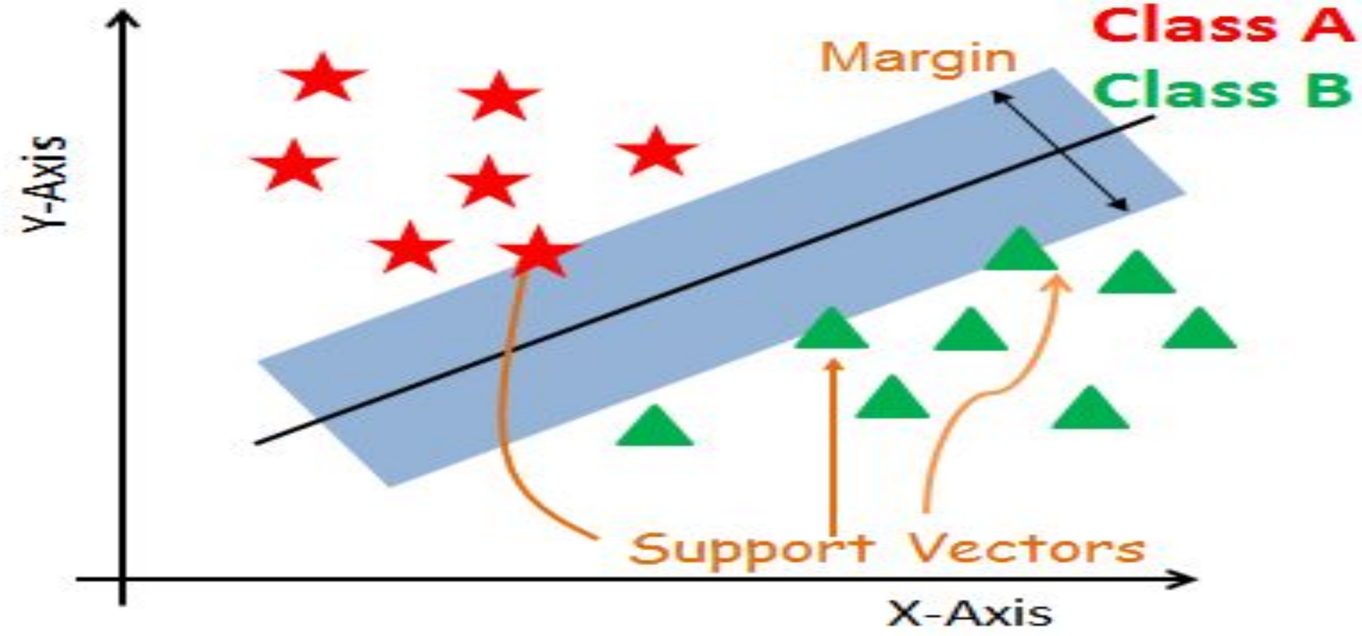
Generally, Support Vector Machines is considered to be a classification approach, but it can be employed in both types of classification and regression problems.

It can easily handle multiple continuous and categorical variables.

SVM constructs a hyperplane in multidimensional space to separate different classes.

SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error.

The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.



Data points from the given classes which helps us to draw the hyperplane with maximal margin are called as support vectors.

Support Vectors

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

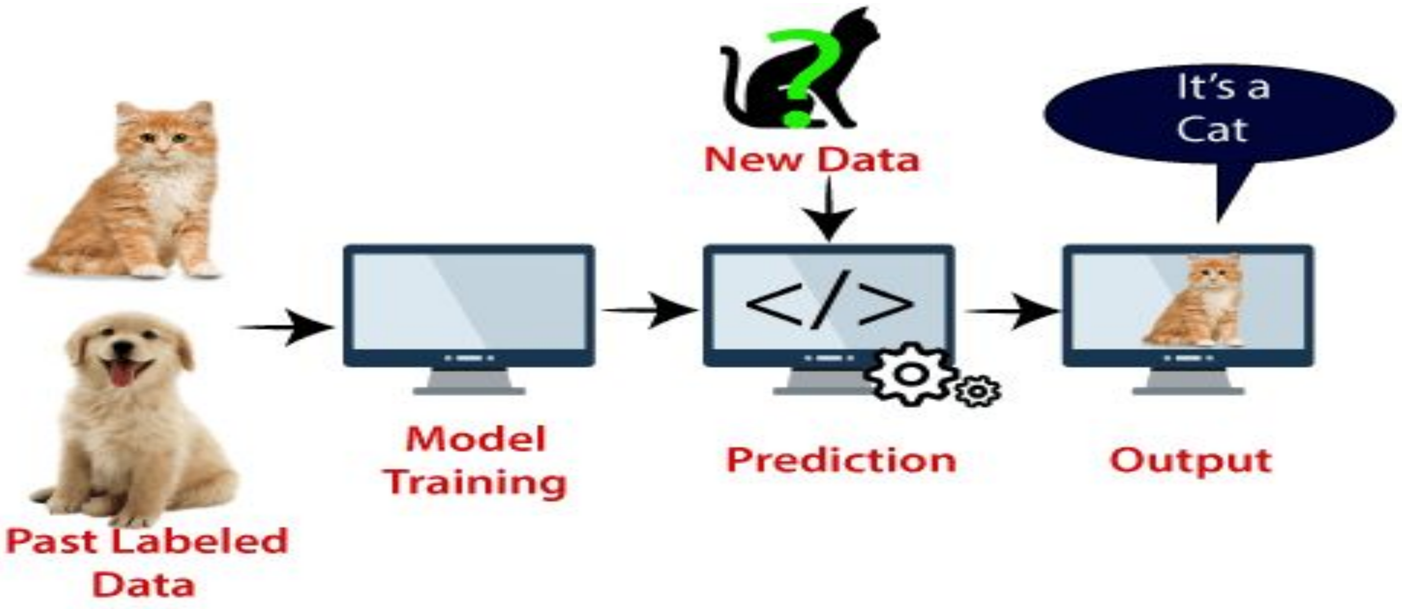
Hyperplane

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Margin

A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

Example: SVM can be understood with the example. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



SVM algorithm can be used for Face detection, image classification, text categorization, etc.

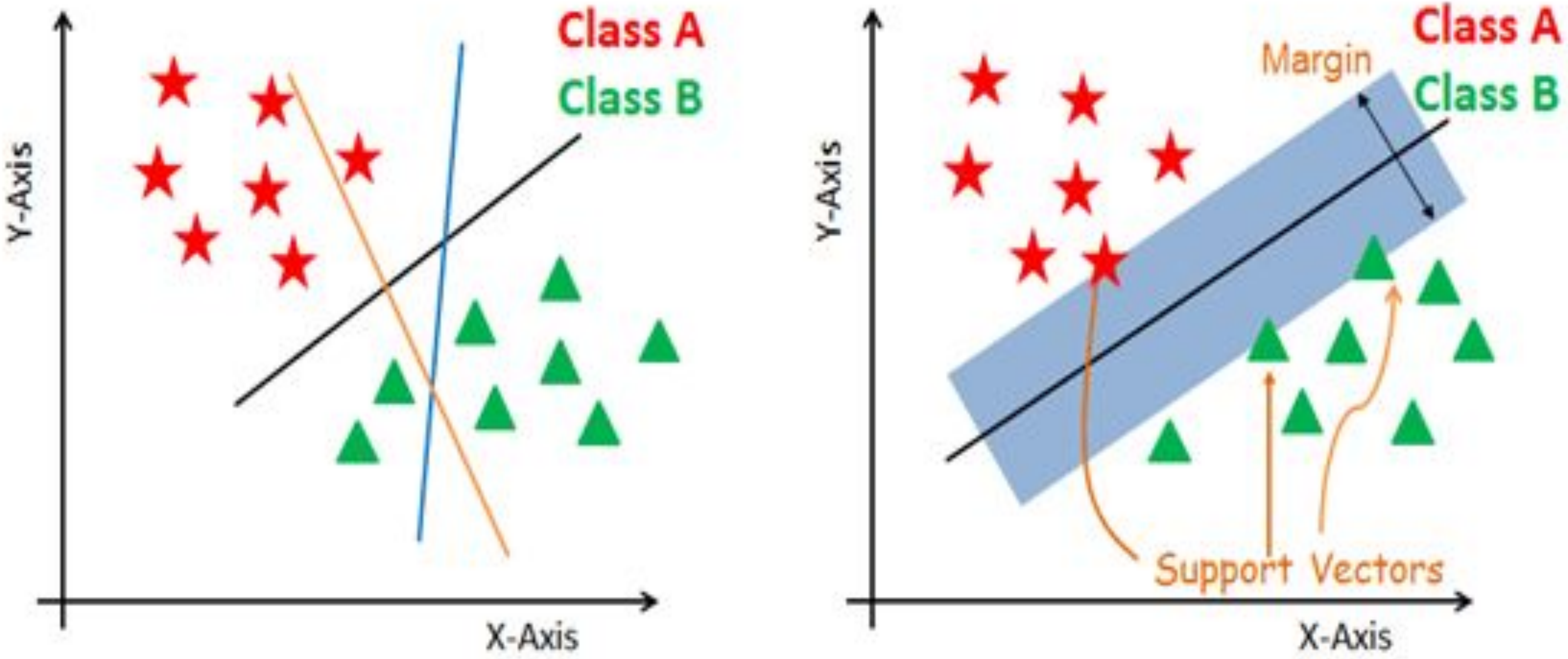
How does SVM work?

- The main objective is to segregate the given dataset in the best possible way.
- The distance between the either nearest points is known as the margin.
- The objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset.

How does SVM work?

SVM searches for the maximum marginal hyperplane in the following steps:

- Generate hyperplanes which segregates the classes in the best way.
- Left-hand side figure showing three hyperplanes **black**, **blue** and **orange.**
- The **blue** and **orange** have higher classification error, but the **black** is separating the two classes correctly.
- Select the right hyperplane with the maximum segregation from the either nearest data points as shown in the right-hand side figure.

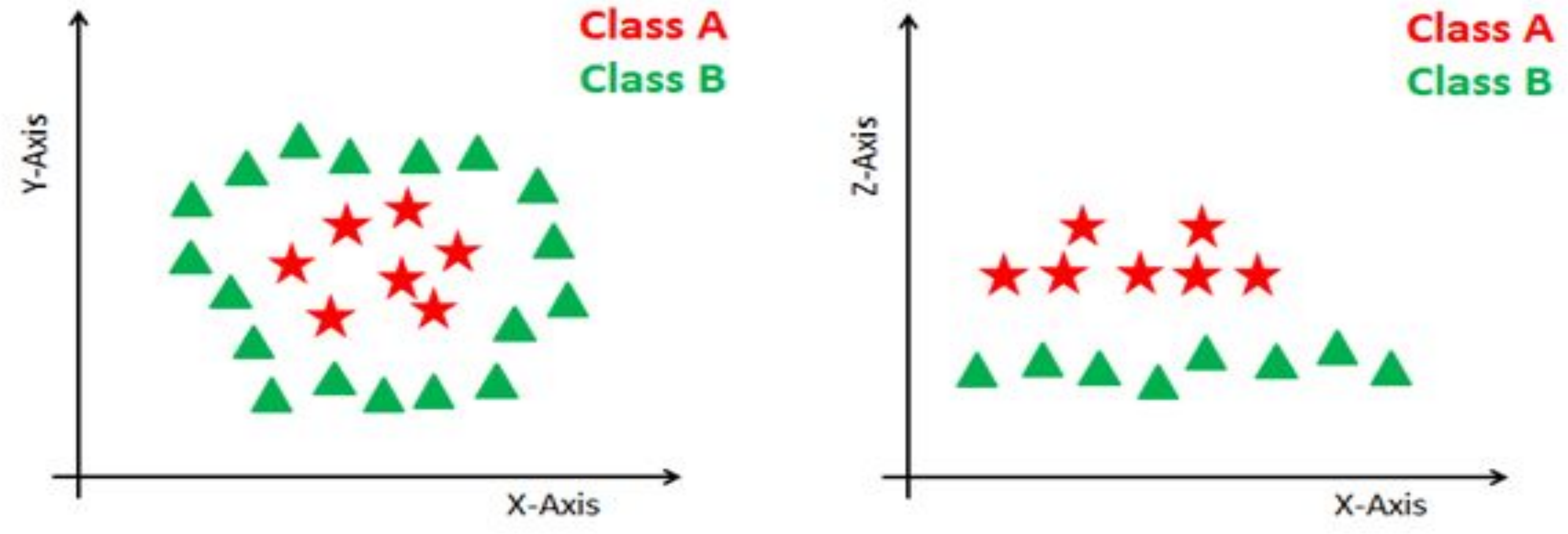

Class A
Class B
Y-Axis
X-Axis
Class A
Class B
Margin
Y-Axis
Support Vectors
X-Axis

Dealing with non-linear and inseparable planes

Some problems can’t be solved using linear hyperplane, as shown in the figure below (left-hand side).

In such situation, SVM uses a **kernel trick** to transform the input space to a higher dimensional space as shown on the right.

The data points are plotted on the x-axis and z-axis (Z is the squared sum of both x and y: **z=x^2+y^2**). Now you can easily segregate these points using linear separation.

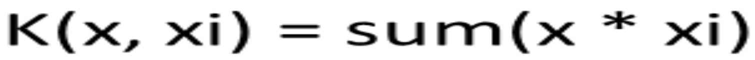


SVM Kernels

- The SVM algorithm is implemented in practice using a kernel.
- A kernel transforms an input data space into the required form.
- SVM uses a technique called the kernel trick.
- The kernel takes a low-dimensional input space and transforms it into a higher dimensional space.
- In other words, you can say that it converts non separable problem to separable problems by adding more dimension to it.
- It is most useful in non-linear separation problem.
- Kernel trick helps you to build a more accurate classifier

SVM Kernels

A Linear Kernel can be used as normal dot product any is the sum of the multiplication of each pair of input values. two given observations. The product between two vectors



$$K(x, xi) = sum(x * xi)$$

Polynomial Kernel

A polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel can distinguish curved or nonlinear input space.

$$K(x,xi) = 1 + sum(x * xi)^d$$

Where,

d is the degree of the polynomial.

d=1 is similar to the linear transformation.

The degree needs to be manually specified in the learning algorithm.

Radial Basis Function Kernel

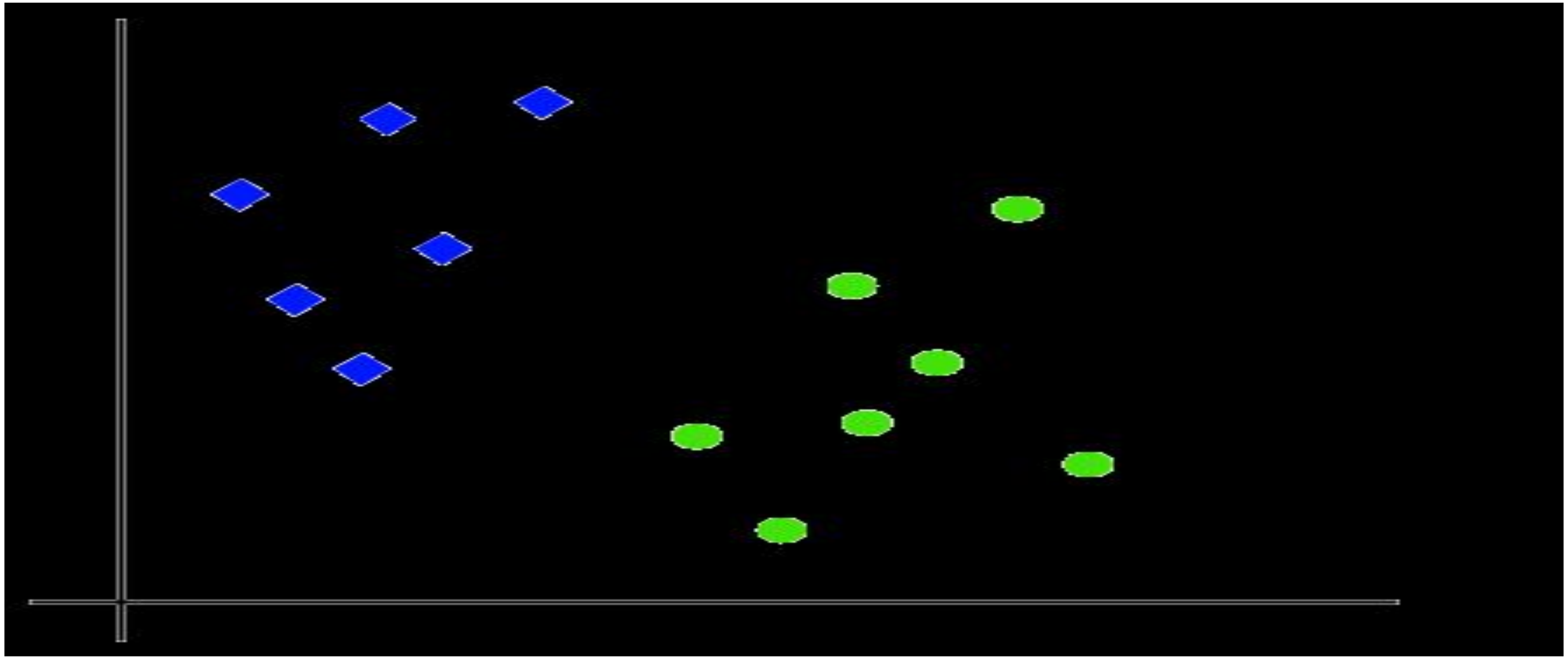
The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification.
RBF can map an input space in infinite dimensional space

$$K(x,xi) = exp(-gamma * sum((x - xi^2))$$

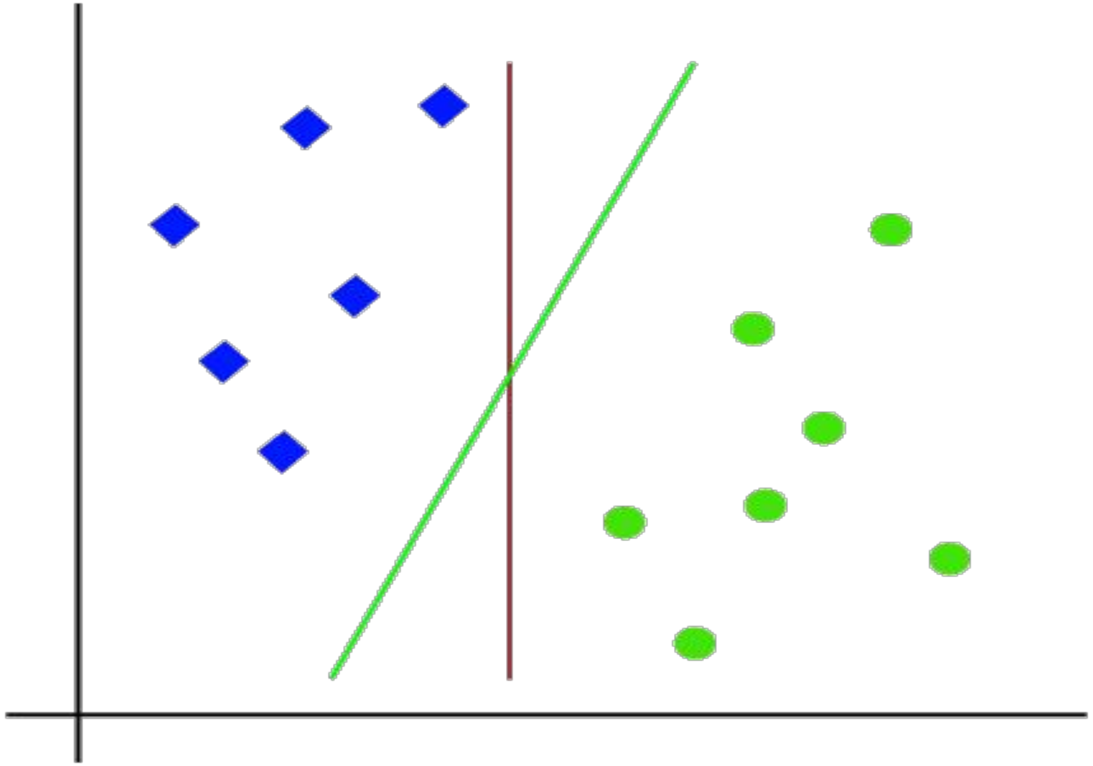
Here **gamma** is a parameter, which ranges from 0 to 1.
A higher value of gamma will perfectly fit the training dataset, which causes over-fitting.
Gamma=0.1 is considered to be a good default value.
The value of gamma needs to be manually specified in the learning algorithm.

Linear SVM

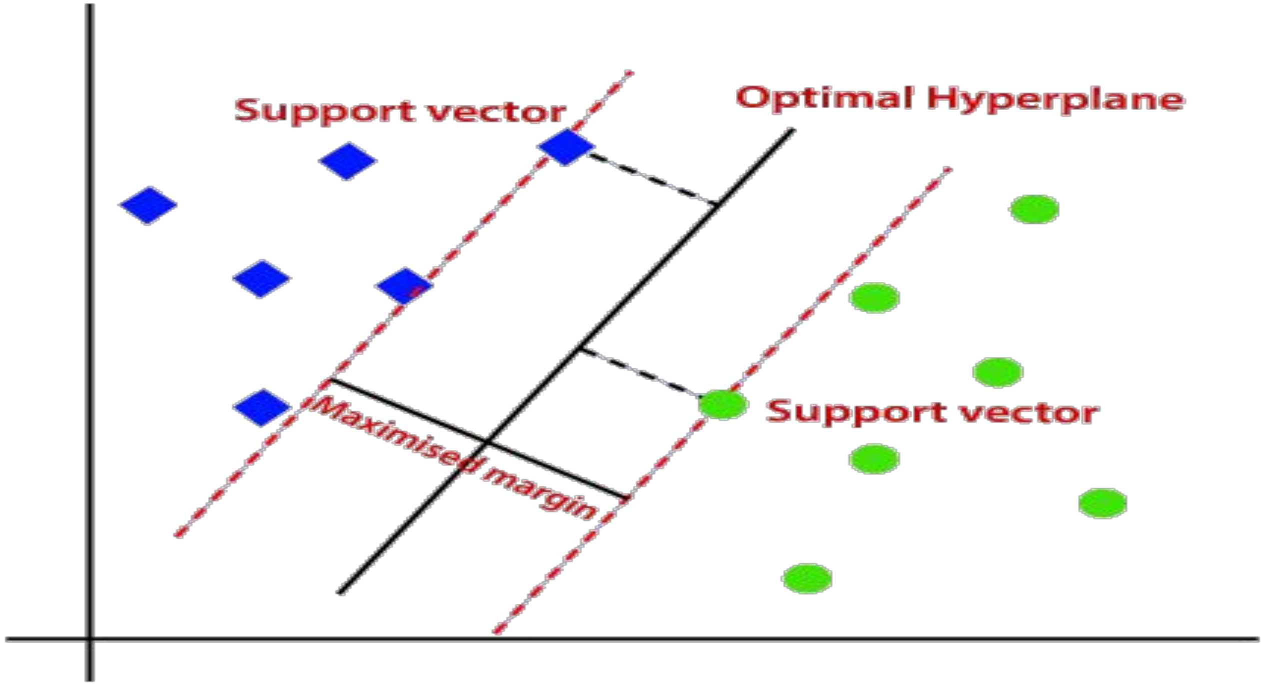
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

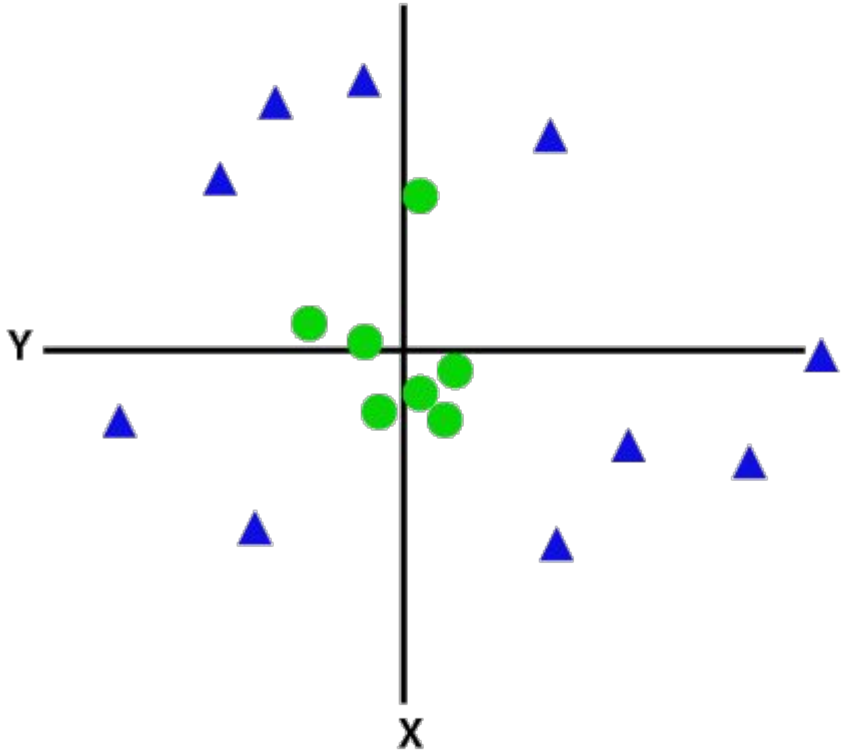


Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.



Non-Linear SVM:

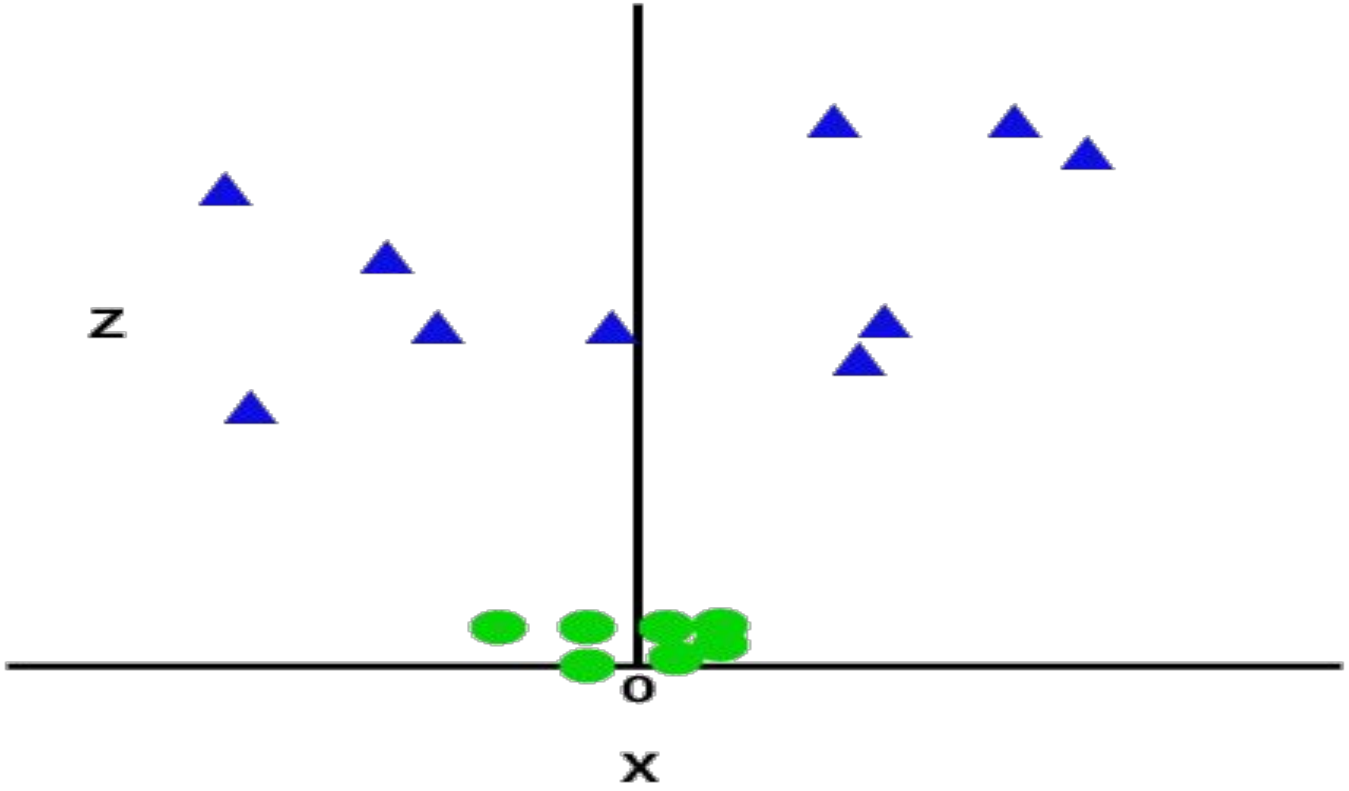
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



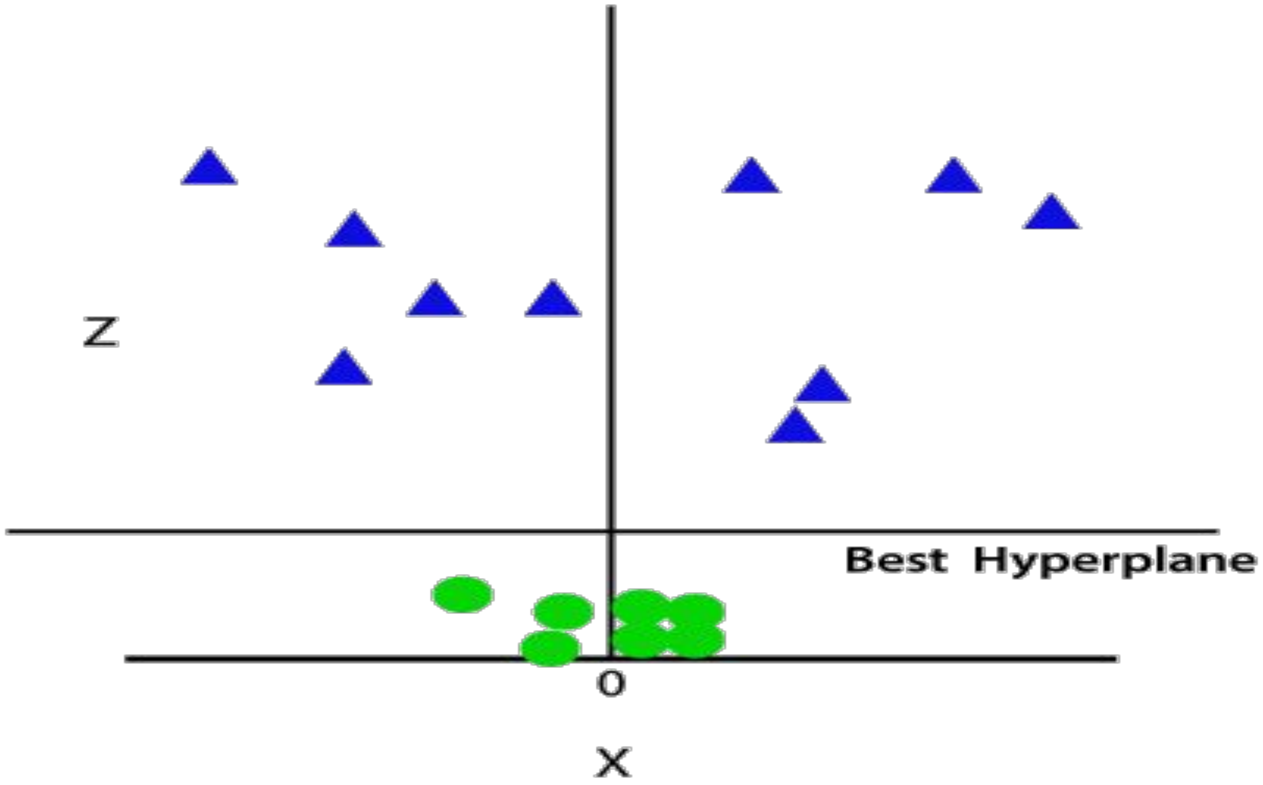
So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

$$z=x2 +y2$$

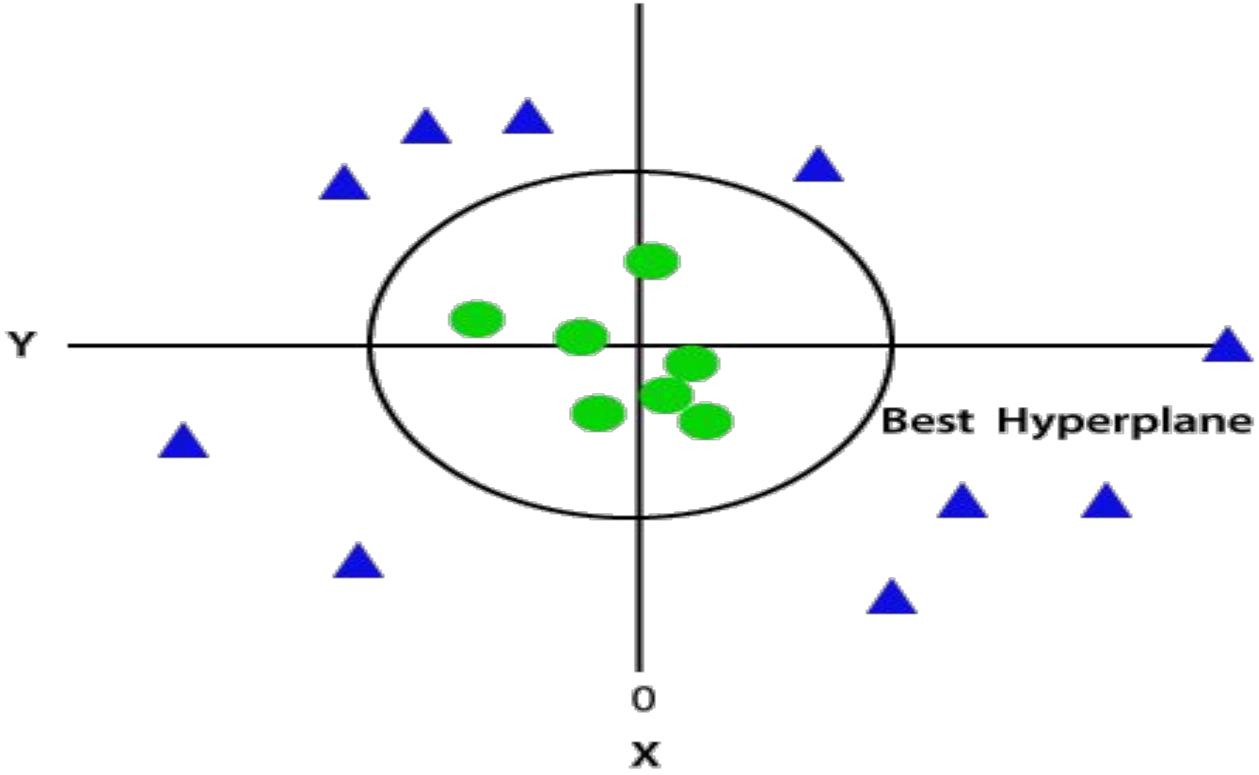
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below imag



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Tuning Hyperparameters

- **Kernel**: The main function of the kernel is to transform the given dataset input data into the required form. There are various types of functions such as linear, polynomial, and radial basis function (RBF). Polynomial and RBF are useful for non-linear hyperplane. Polynomial and RBF kernels compute the separation line in the higher dimension. In some of the applications, it is suggested to use a more complex kernel to separate the classes that are curved or nonlinear. This transformation can lead to more accurate classifiers.
- **Regularization**: Regularization parameter in python's Scikit-learn C parameter used to maintain regularization. Here C is the penalty parameter, which represents misclassification or error term. The misclassification or error term tells the SVM optimization how much error is bearable. This is how you can control the trade-off between decision boundary and misclassification term. A smaller value of C creates a small-margin hyperplane and a larger value of C creates a larger-margin hyperplane.
- **Gamma**: A lower value of Gamma will loosely fit the training dataset, whereas a higher value of gamma will exactly fit the training dataset, which causes over-fitting. In other words, you can say a low value of gamma considers only nearby points in calculating the separation line, while the a value of gamma considers all the data points in the calculation of the separation line.

Advantages

- SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm.
- They also use less memory because they use a subset of training points in the decision phase.
- SVM works well with a clear margin of separation and with high dimensional space.

Disadvantages

- SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes.
- It works poorly with overlapping classes and is also sensitive to the type of kernel used.