SNS COLLEGE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTION)

Approved by AICTE & Affiliated to Anna University
Accredited by NBA & Accrediated by NAAC with 'A+' Grade,
Recognized by UGC saravanampatti (post), Coimbatore-641035.

Department of Artificial Intelligence and Machine learning

Course Name: 19AMT301- Deep Learning

III Year : VI Semester

Unit I –INTRODUCTION

Topic : Support Vector Machine

SVM

- SVM is a powerful supervised algorithm that works well with smaller data set but with complex one
- Used for both regression and classification but preferred as a classifier
- Famous since 1990s and more effective with tuning

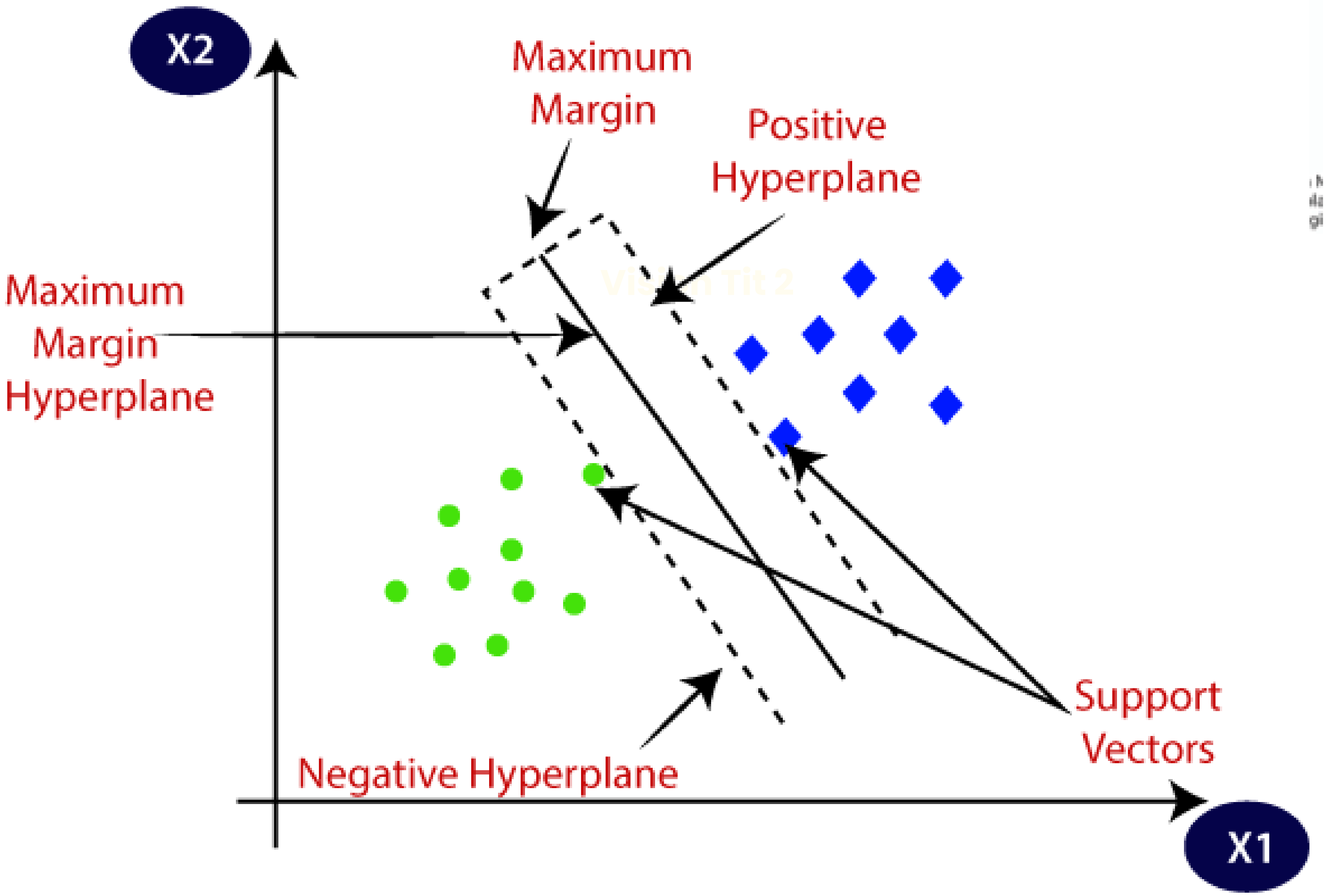
How SVM works?

- It is a supervised machine learning problem where we try to find a hyperplane that best separates the two classes
- Logistic regression Vs SVM- Both are similar without Kernel but with Kernel SVM is preferred
- For Small and complex dataset, go-to is SVM

SVM Types

- Linear SVM- Perfectly linearly separable data, can be classified as 2 classes (2D)
- Non-Linear SVM- NOT linearly separable data, kernel tricks needed, 2 Classes not possible (2D)

SVM graph



X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

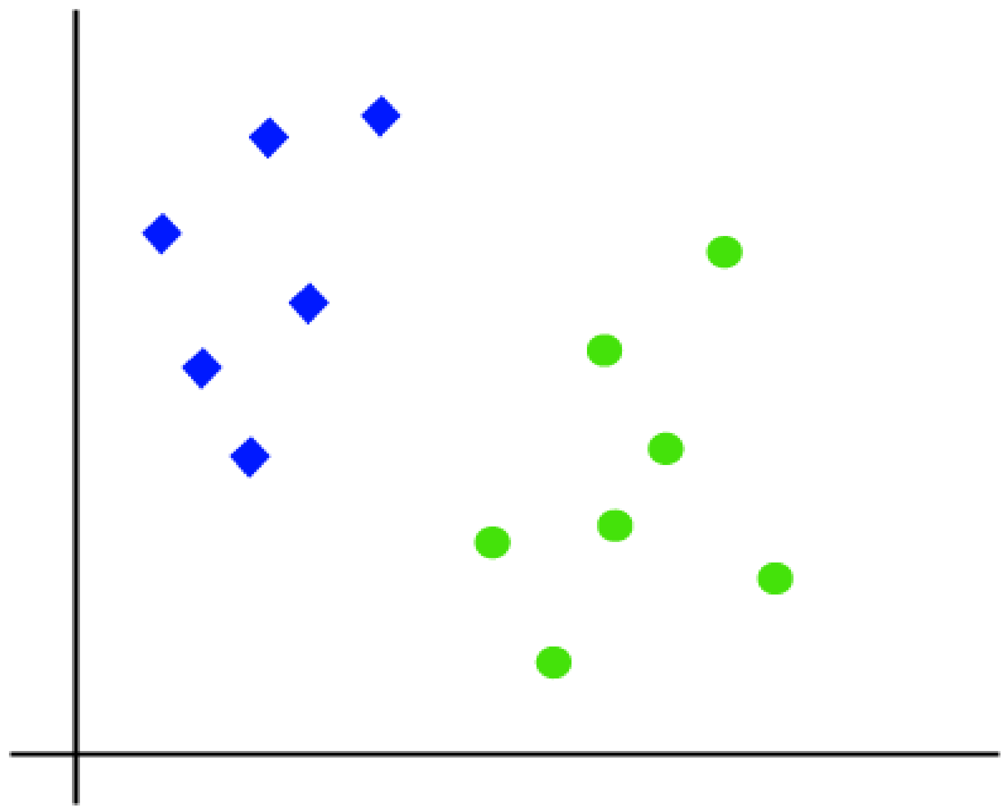
Important terms in SVM

- 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.**

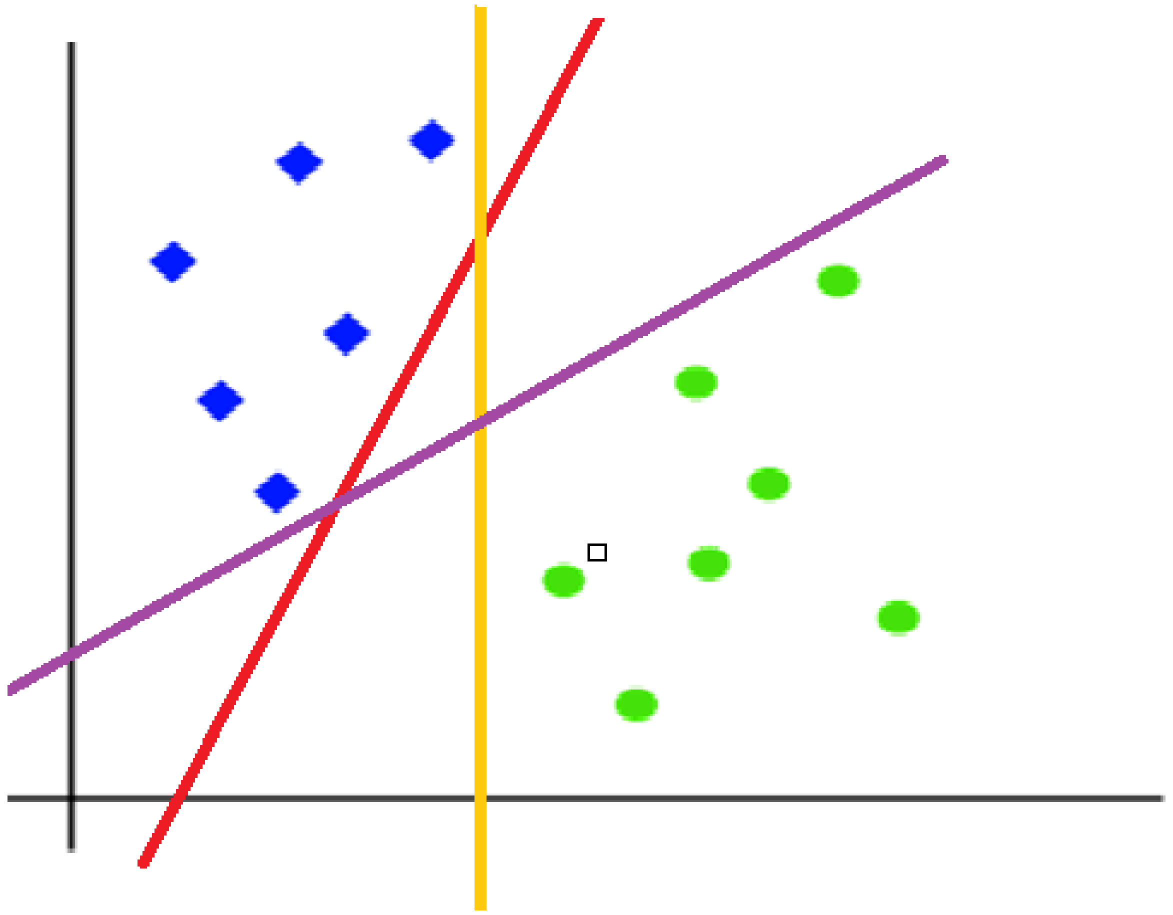
SVM working with example

- Lets take an example that the dataset has two classes(Blue and Green). Now we want to classify whether the new dataset is either Blue or Green
- 2D- Decision boundary or straight line or Separating line
- 3D- Hyperplane

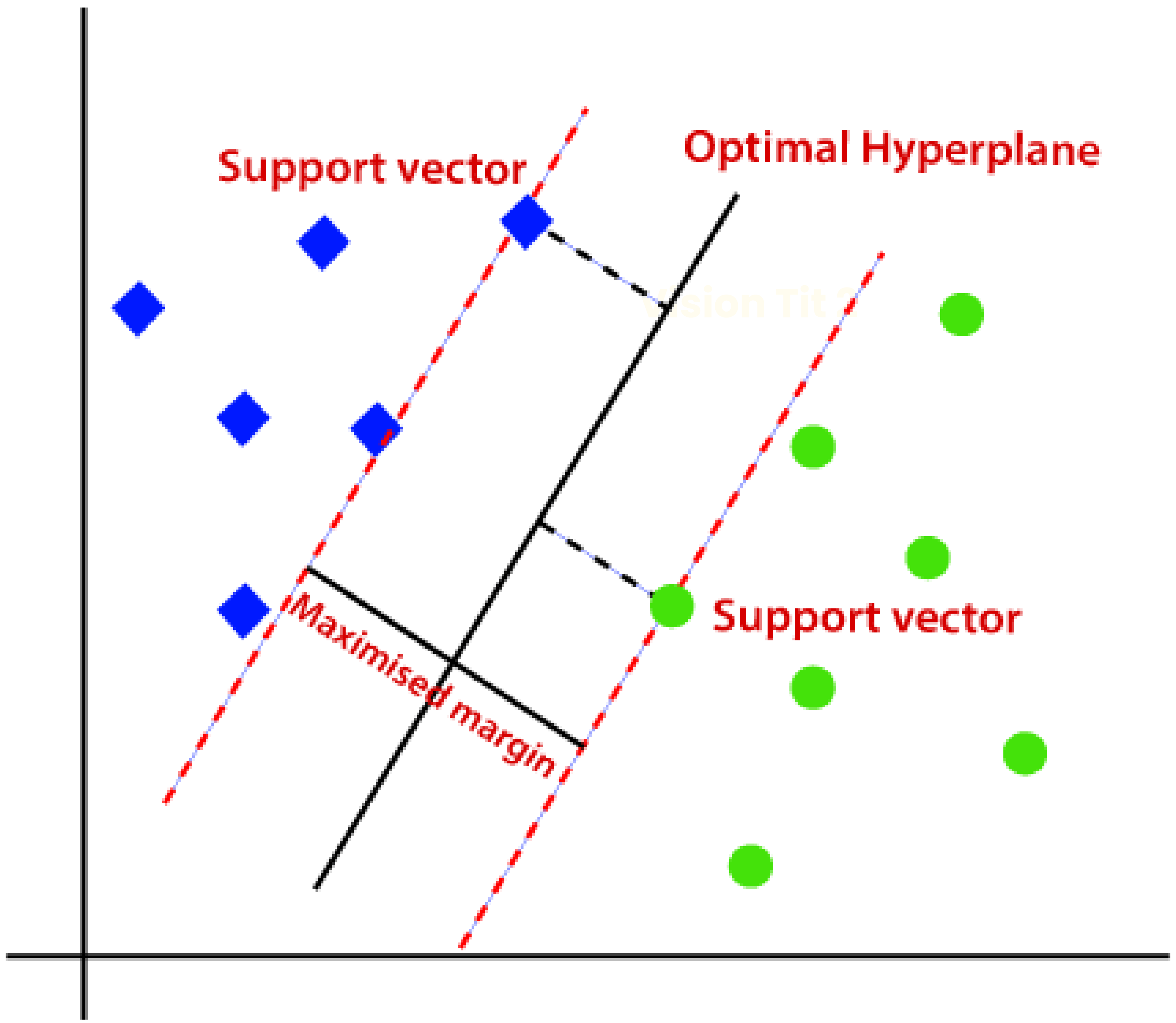


Selection of Hyperplane

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.



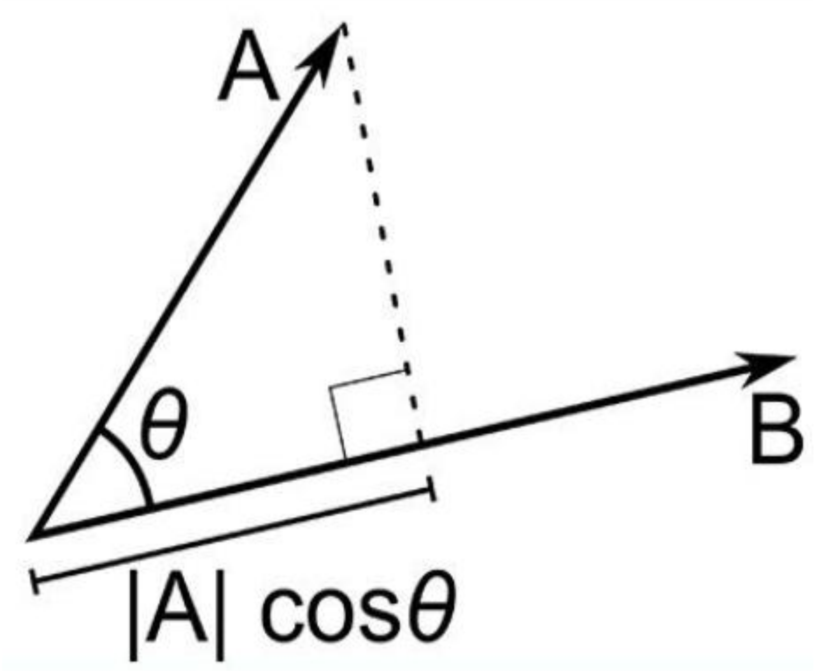
Optimal Hyperplane



Mathematical intuition behind SVM

Dot Product:

The dot product can be defined as the projection of one vector along with another, multiply by the product of another vector.

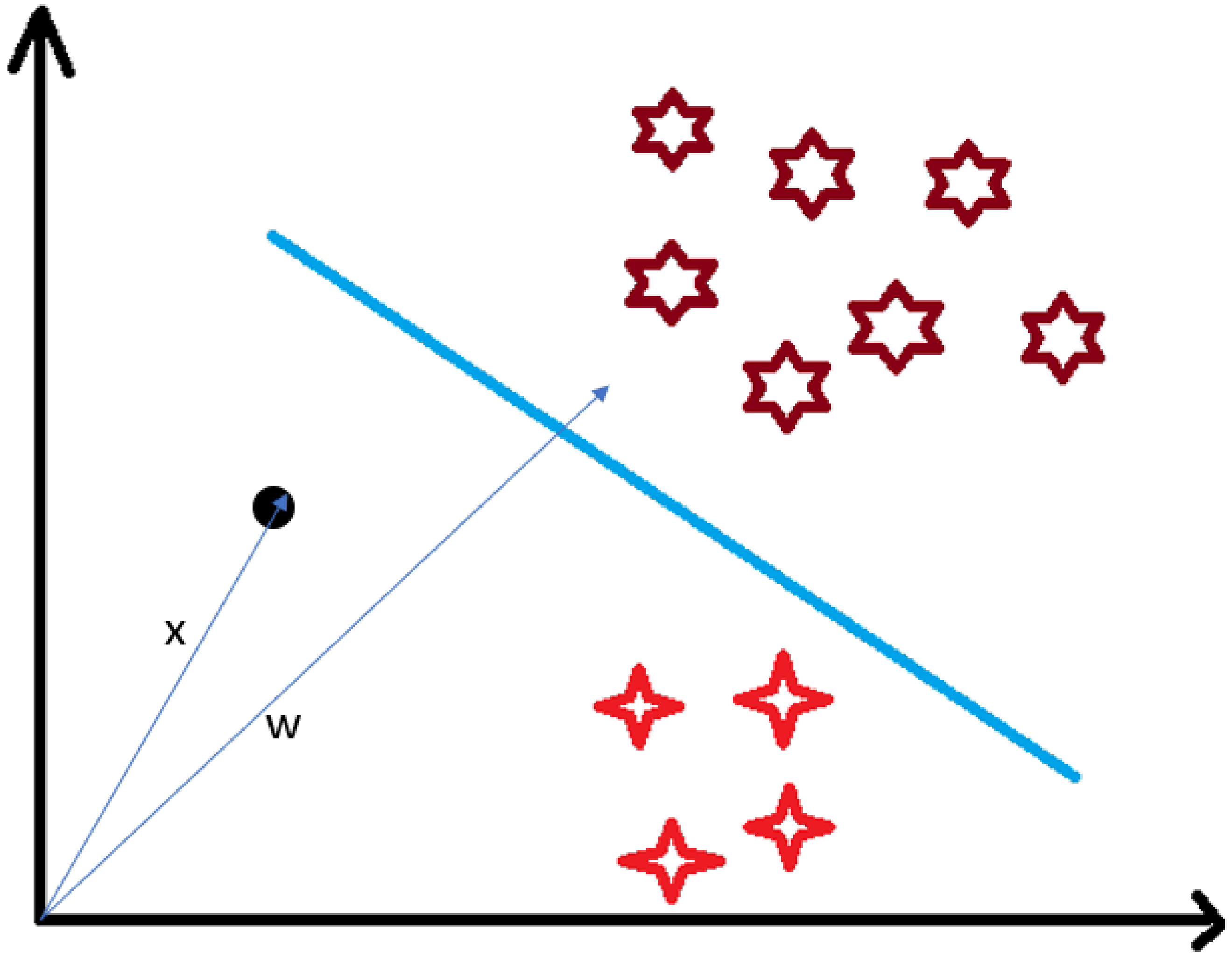


A . B = |A| cosθ * |B|

A.B = |A| cosθ * unit vector of B

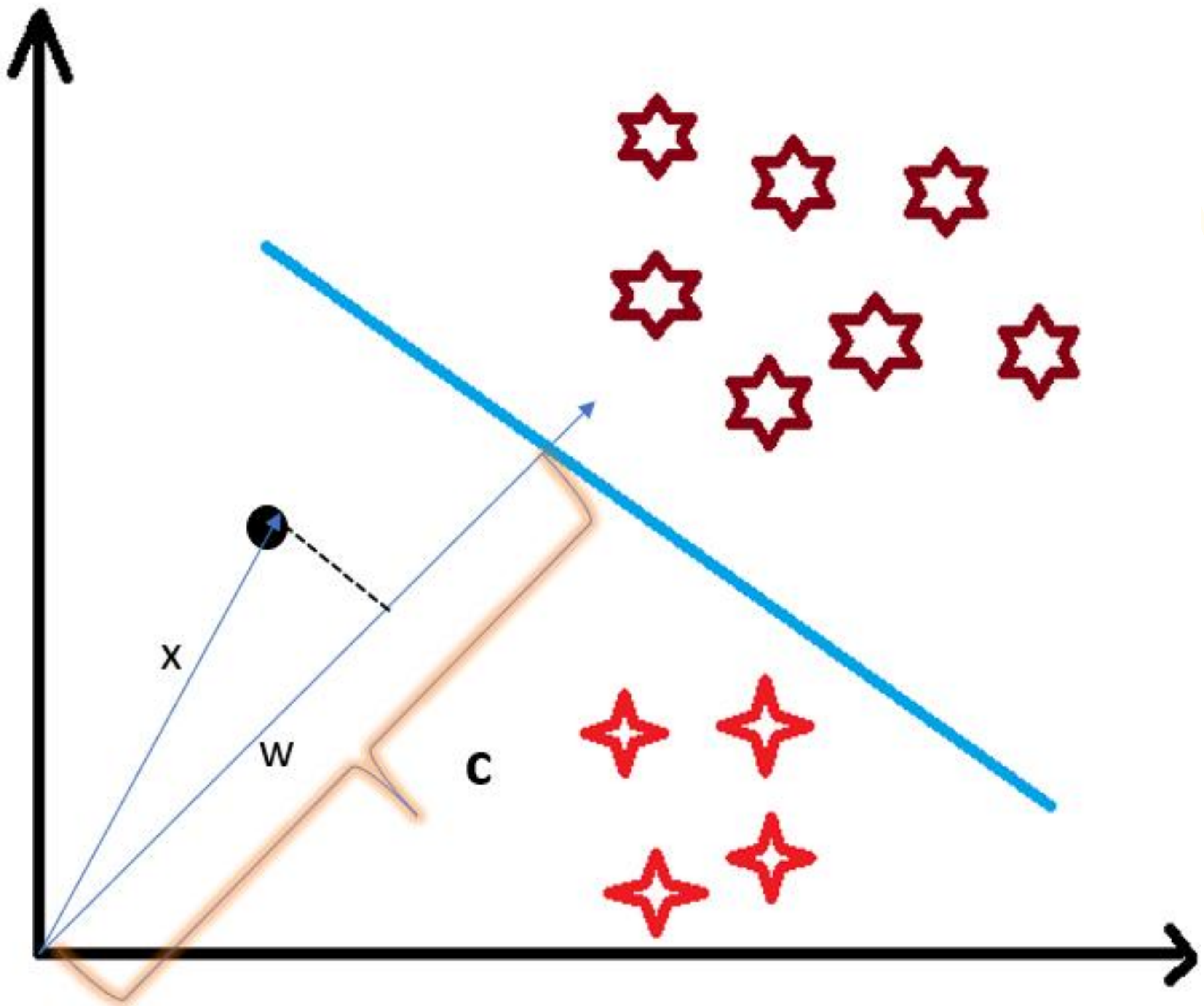
Use of Dot product in SVM

Consider a random point X and we want to know whether it lies on the right side of the plane or the left side of the plane (positive or negative).



Projection of X on W

"C" is distance of vector w from origin to decision boundary



Dot product of X and W

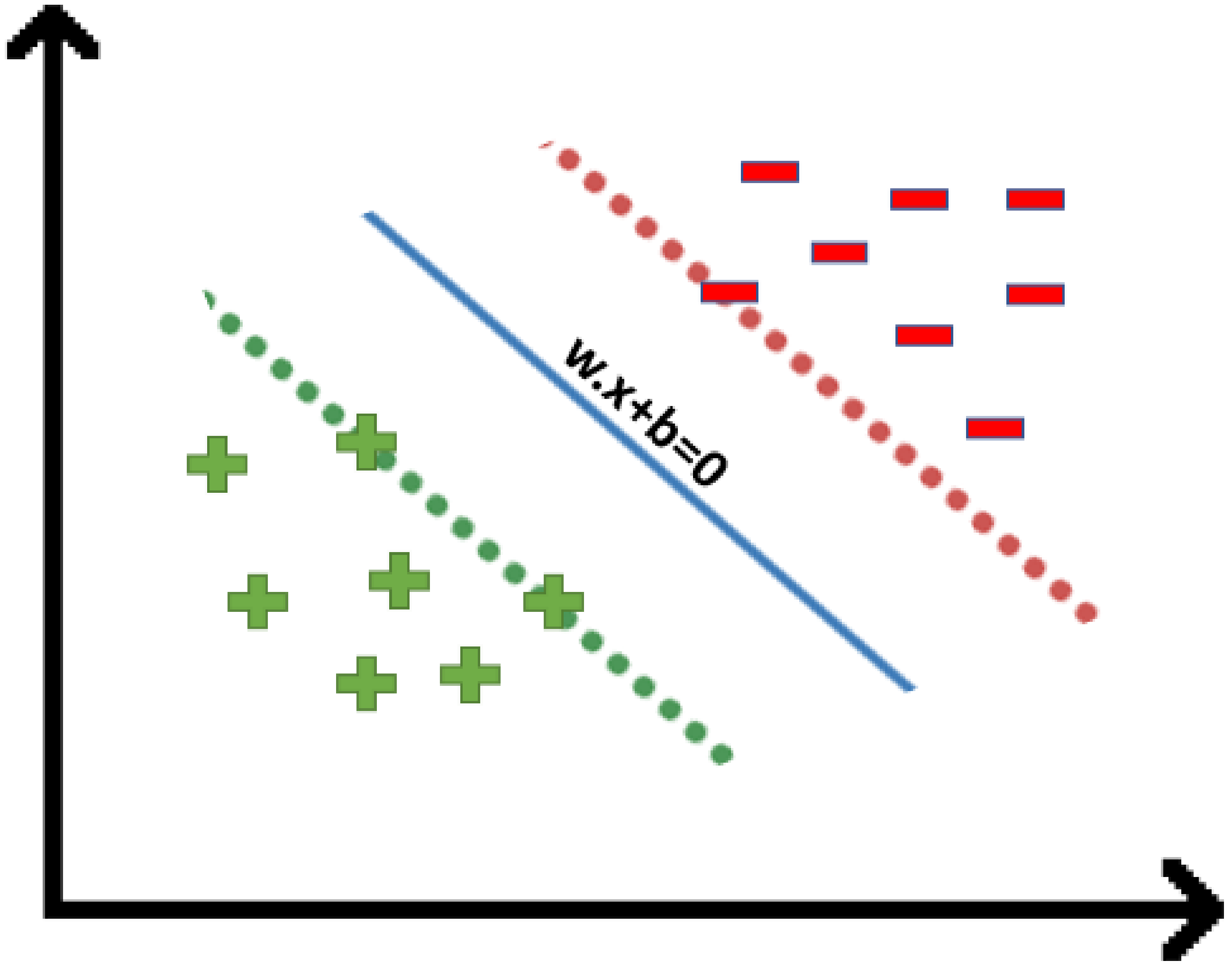
$$\vec{X}.\vec{w} = c \;\; (\textit{the point lies on the decision boundary})$$

$$\vec{X}.\vec{w} > c \; (\textit{positive samples})$$

$$\vec{X}.\vec{w} < c \; (\textit{negative samples})$$

Margin in SVM

- The equation of a hyperplane is w.x+b=0 where w is a vector normal to hyperplane and b is an offset.



Margin in SVM

Decision Rule:

$$\vec{X}.\vec{w} - c \geq 0$$

$$\text{putting } -c \text{ as b, we get}$$

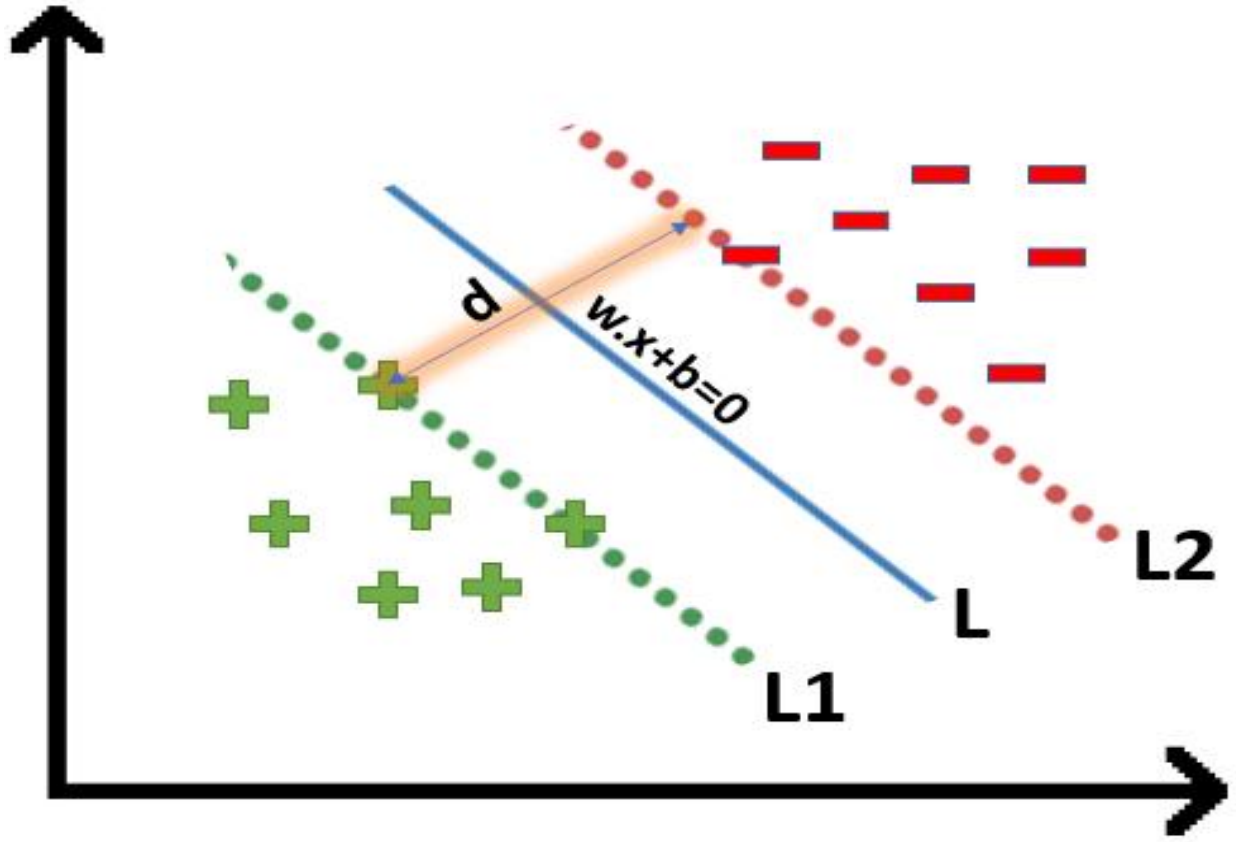
$$\vec{X}.\vec{w} + b \geq 0$$

$$\text{hence}$$

$$y = \begin{cases} +1 & \text{if } \vec{X}.\vec{w} + b \geq 0 \\ -1 & \text{if } \vec{X}.\vec{w} + b < 0 \end{cases}$$

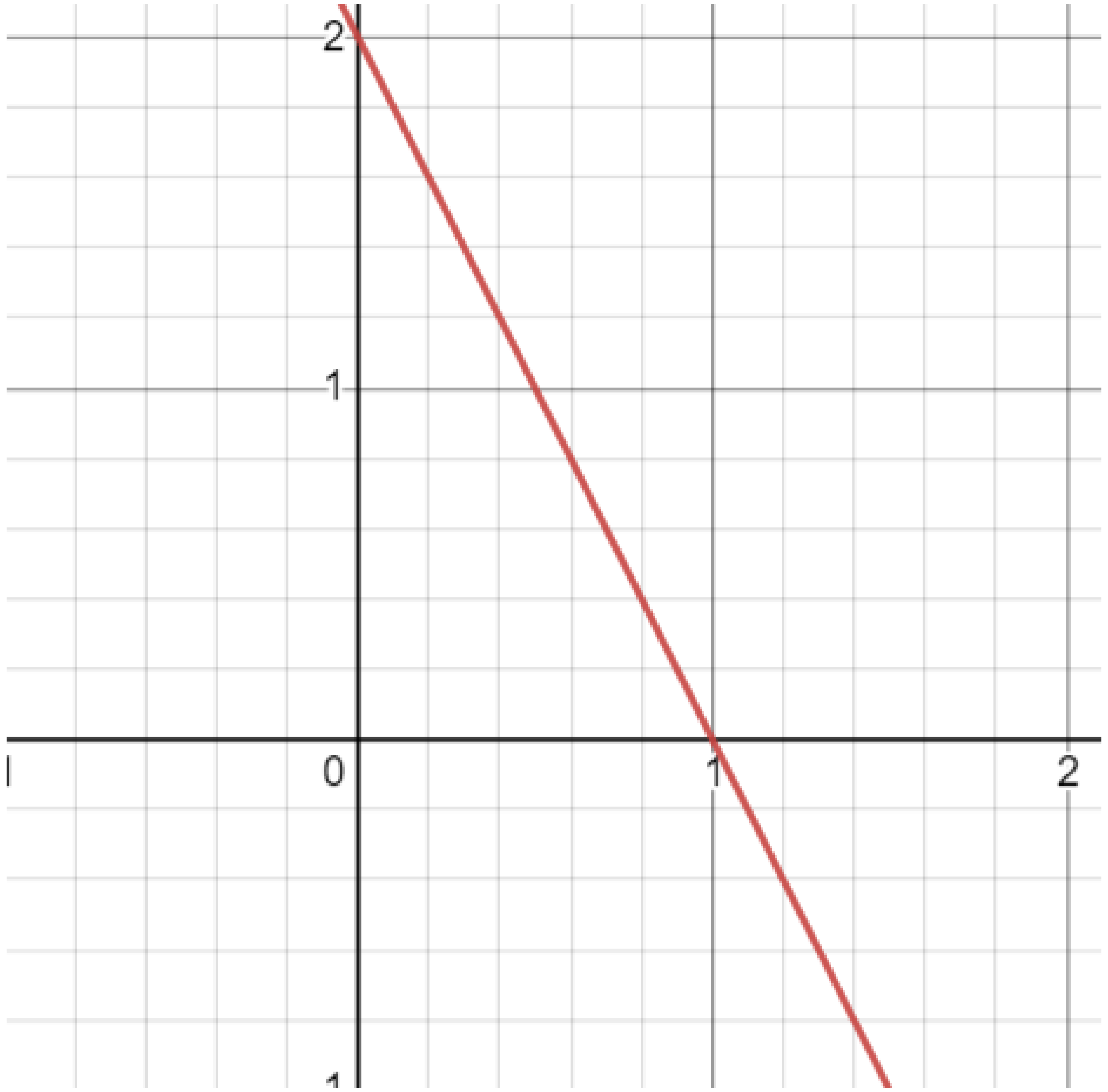
Calculate W and d such that margin has a maximum distance

To calculate 'd' we need the equation of L1 and L2. For this, we will take few assumptions that the equation of L1 is **w.x+b=1** and for L2 it is **w.x+b=-1**.

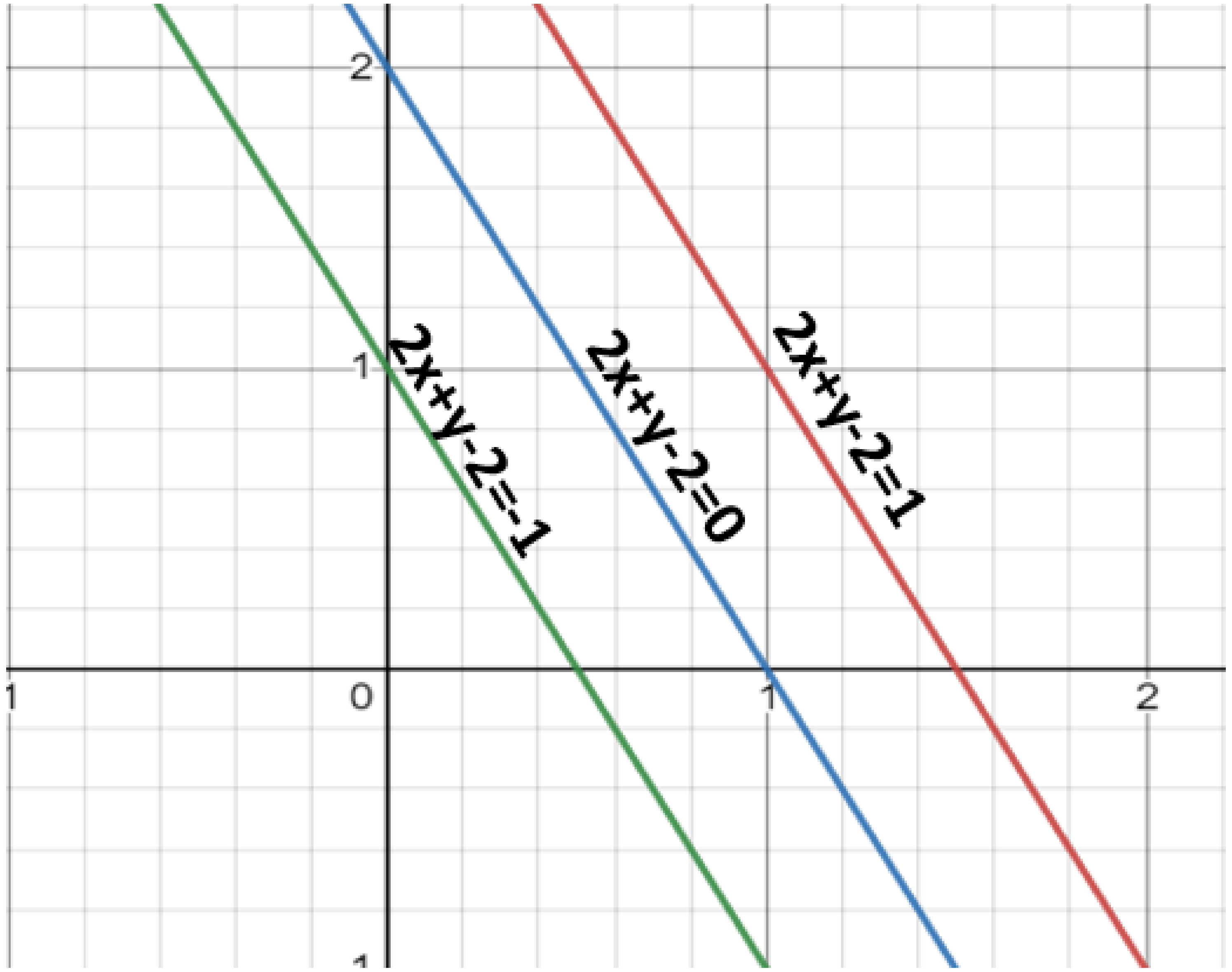


Calculate W and d such that margin has a maximum distance

Let equation of hyperplane be 2x+y=2



Margin for the hyperplane



Optimization and its constraints

- In order to get our optimization function, there are few constraints to consider. That constraint is that **“We’ll calculate the distance (d) in such a way that no positive or negative point can cross the margin line”.** Let’s write these constraints mathematically:

$$For\ all\ the\ Red\ points\ \vec{w}.\vec{X}+b \leq -1$$

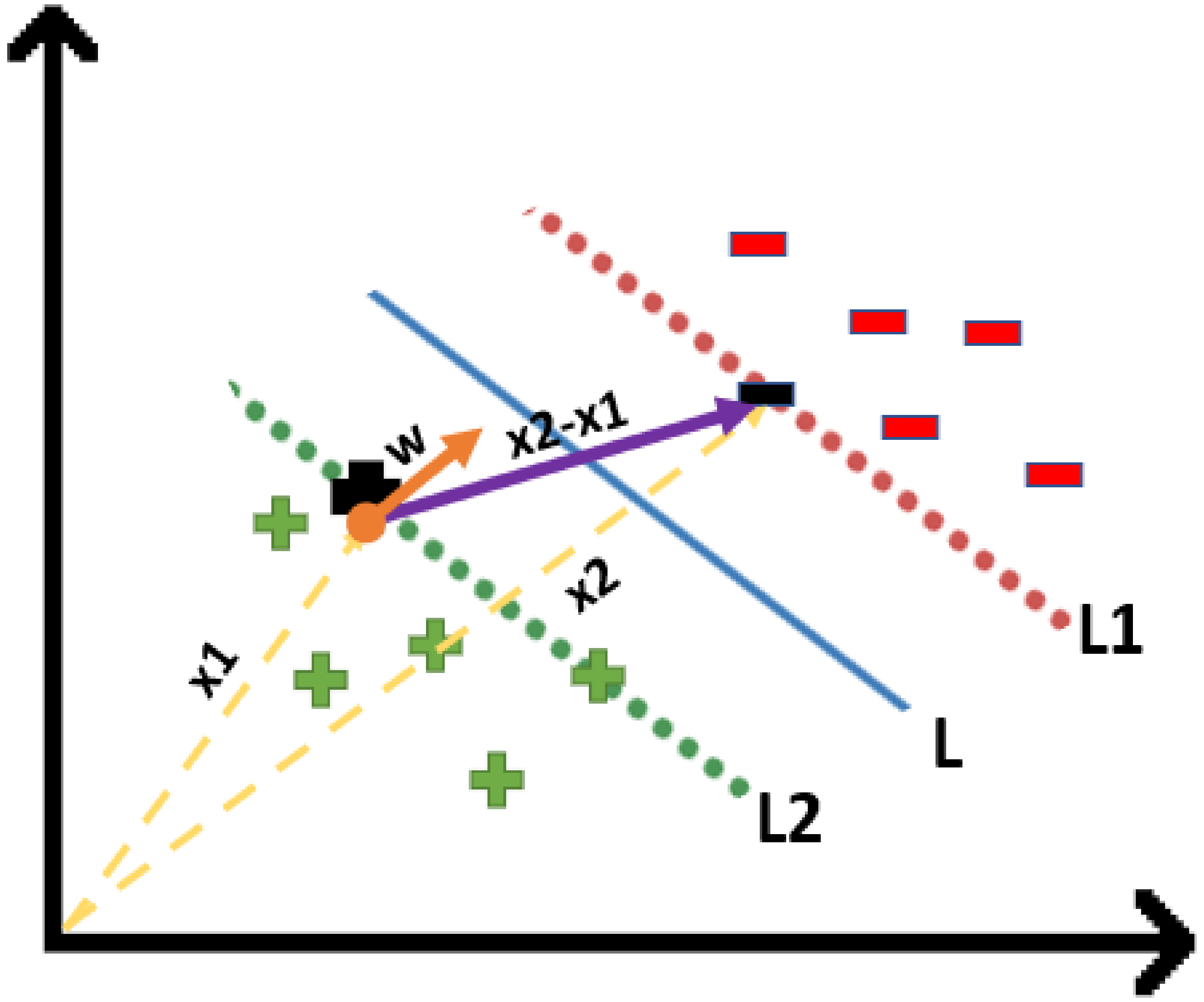
$$For\ all\ the\ Green\ points\ \vec{w}.\vec{X}+b \geq 1$$

Condition for correct classification

- Lets make
- Y=1 for positive class
- Y=-1 for negative class
- Condition for correct classification of every point is,

$$y_i(\vec{w}.\vec{X}+b) \geq 1$$

Projection of w on x2-x1



$$\Rightarrow (x2 - x1) \cdot \frac{\vec{w}}{\|w\|}$$

$$\Rightarrow \frac{x2 \cdot \vec{w} - x1 \cdot \vec{w}}{\|w\|} \quad ----(1)$$

- Since x2 and x1 are support vectors and they lie on the hyperplane, hence they will follow **y_i* (2.x+b)=1** so we can write it as:

for positive point $y = 1$

$$\Rightarrow 1 \times (\vec{w} . x1 + b) = 1$$

$$\Rightarrow \vec{w} . x1 = 1 - b \qquad ------(2)$$

Similarly for negative point $y = -1$

$$\Rightarrow -1 \times \left(\vec{w} . x2 + b\right) = 1$$

$$\Rightarrow \vec{w} . x2 = -b - 1 \qquad ------(3)$$

- Putting equations (2) and (3) in equation (1) we get:

$$\Rightarrow \frac{(1-b)-(-b-1)}{\|w\|}$$

$$\Rightarrow \frac{1-b+b+1}{\|w\|} = \frac{2}{\|w\|} = d$$

Hence the equation which we have to maximize is:

$$\operatorname{argmax}(w^*, b^*)\ \frac{2}{\|w\|}\ \text{such that}\ y_i\left(\vec{w}.\vec{X}+b\right)\geq 1$$

Soft Margin

For linearly non seperable data we modify the equation for Soft Margin as,

$$\operatorname{argmin}(w^*, b^*)\ \frac{\|w\|}{2}\ \text{such that}\ y_i\left(\vec{w}.\vec{X}+b\right)\geq 1$$

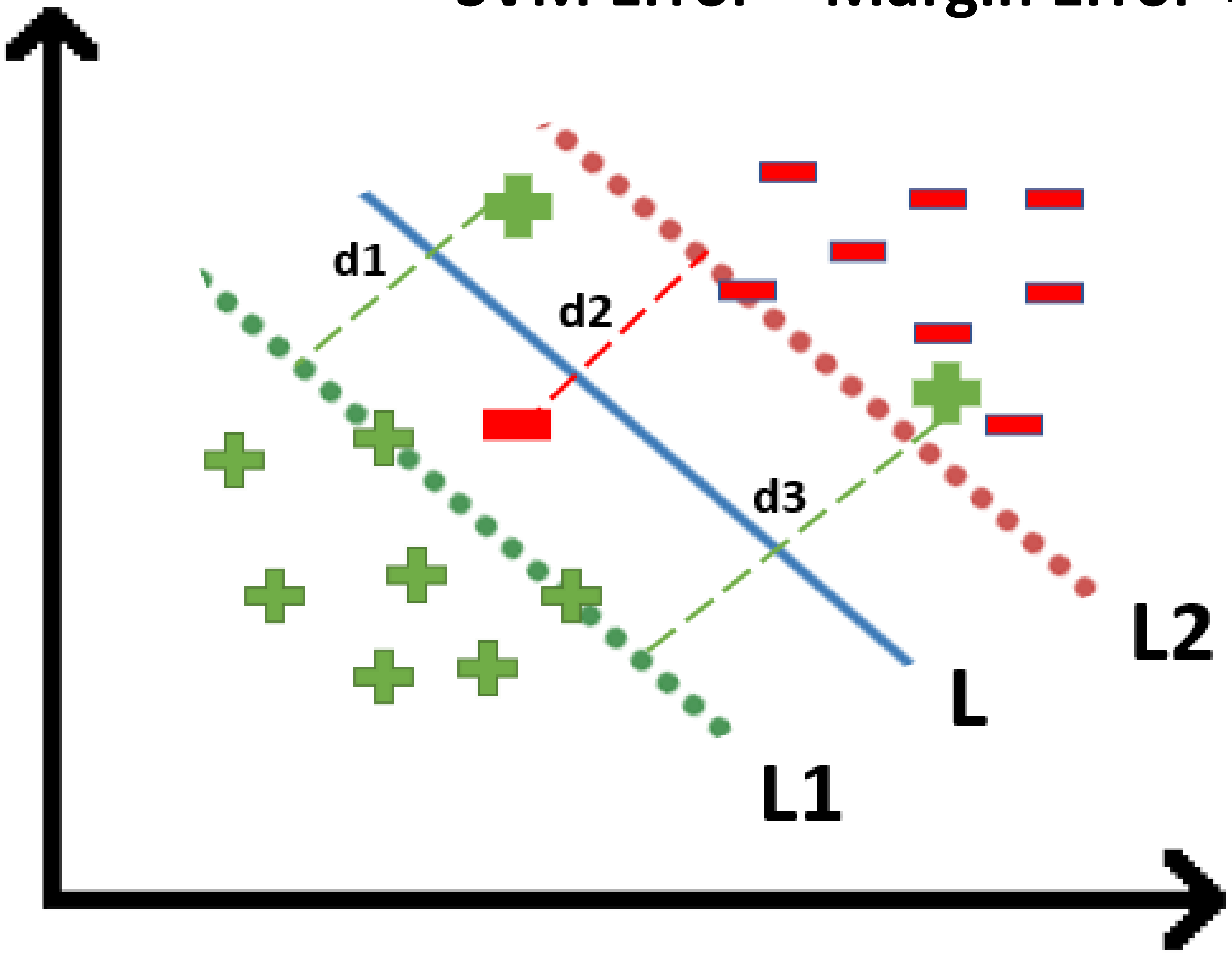
To make a soft margin equation we add 2 more terms to this equation which is **zeta** and multiply that by a **hyperparameter ‘c’**

$$\operatorname{argmin}\left(w^{*}, b^{*}\right) \frac{\|w\|}{2} + c \sum_{i=1}^{n} \zeta_i$$

SVM Error

d1,d2,d3 are zeta value for every misclassification and “0” for correct classification

SVM Error = Margin Error + Classification Error.



Which Model is the best model?

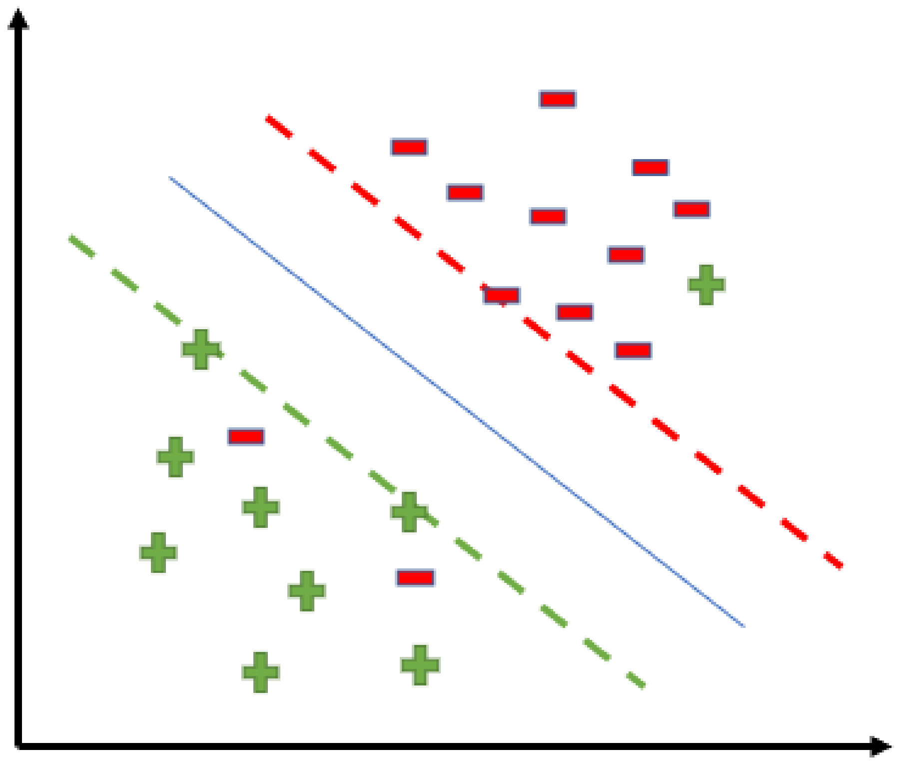


Figure 1

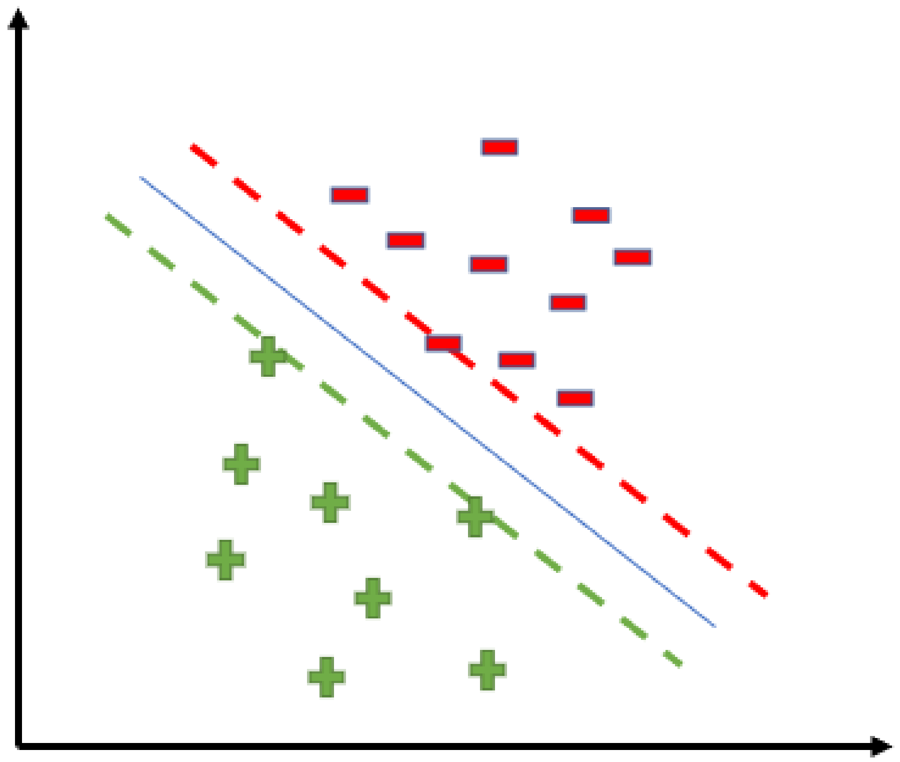
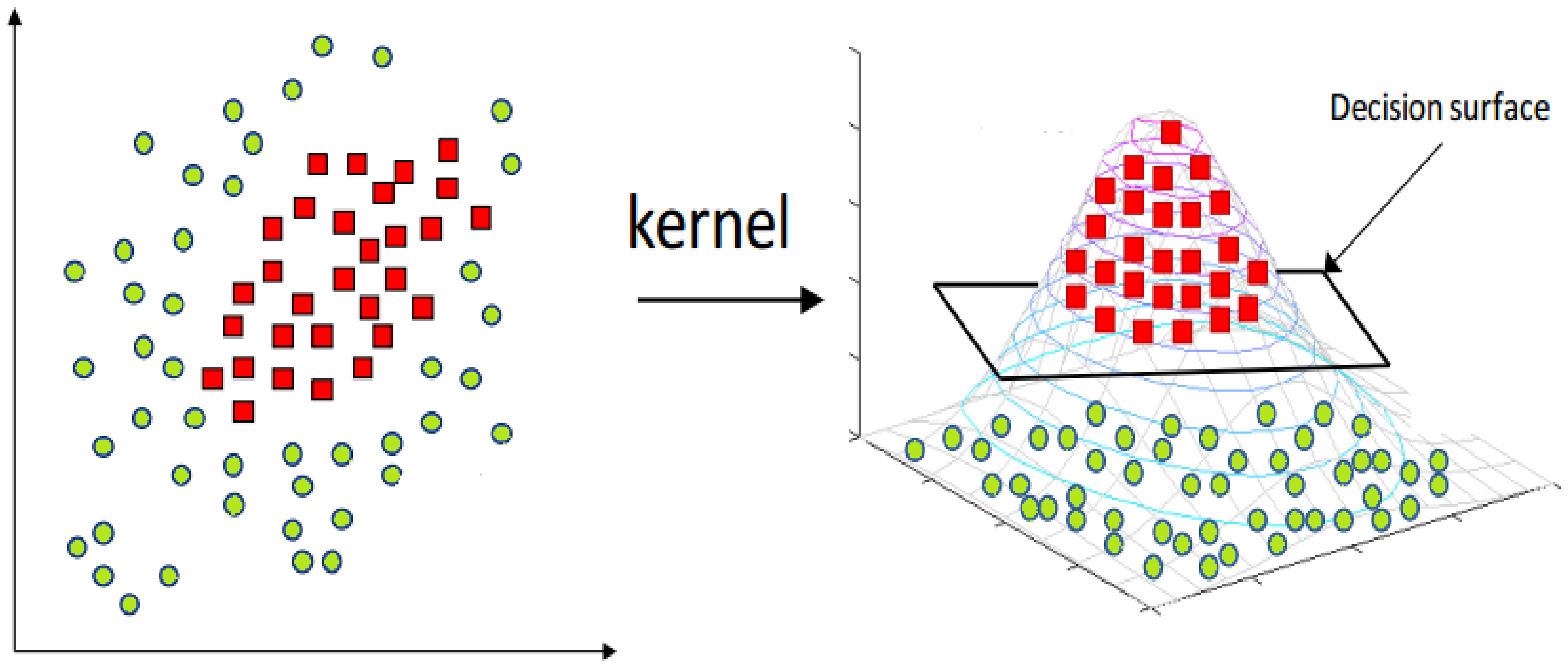


Figure 2

Kernels



THANK YOU !!!