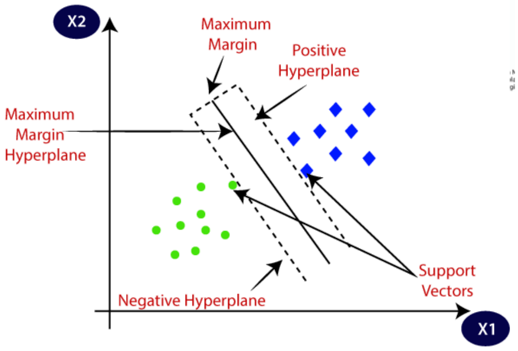
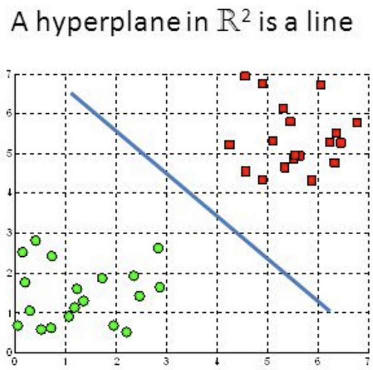
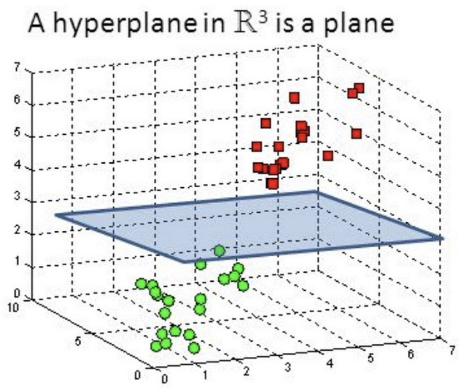



# Understanding Support Vector Machines

## Table Of Contents

# ⊛ Support Vectors Machine (SVM)

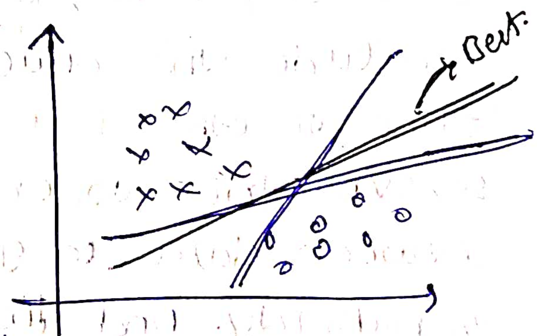
→ Support vector machine is a supervised machine learning algorithm, which can be a classifier as well as Regressor. This is a linear model, and if consider linear classification models we have

→ Logistic Regression and support vector classifiers.

→ In case of logistic Regression, we ultimately build a line/plane/hyperplane and classify the points to the sides of the line, and given a new point based on the probability, we can say which part of plane it belongs. Now, what about SVM.

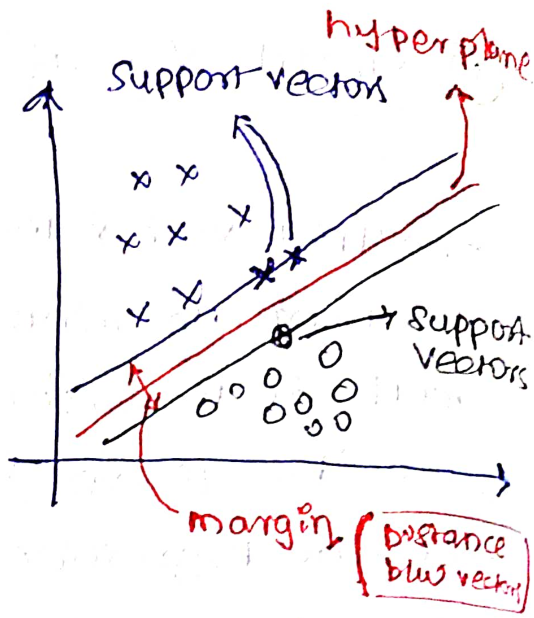
## ⊙ Concept of SVM

→ Similar to logistic Regression, we get a plane/hyper plane in SVM, but how do we decide on the best hyperplane. There can be several planes passing through the data points.




→ So, the key principle behind SVM is to find the hyperplane/decision boundary, that maximally seperates data points belonging to distinct classes, that is, maximizing the margin between datapoints.

## ⊙ what is a support vector



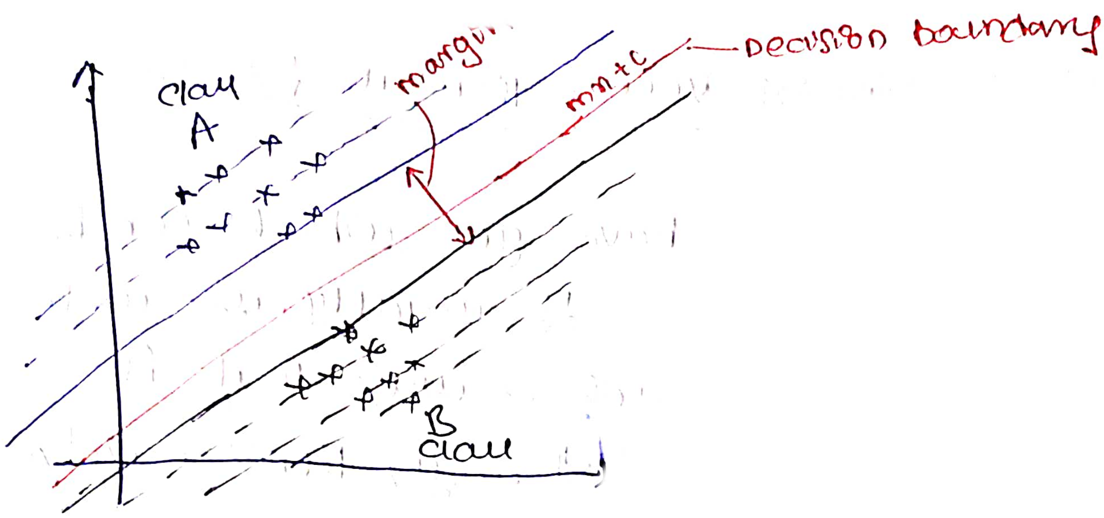

→ let's suppose we have a decision boundary and to both sides of line we take the closest points and build a parallel plane to our decision boundary.

→ the points that are close to decision boundary are called "support vectors". And these support vectors are crucial for defining hyperplane so these support vectors help us find optimal plane

## ⊙ what is margin

→ If we consider logistic Regression, based on the Probability we will decide the class for a new point, but there are chances for things to go wrong here, say probability is 0.49 & so we conclude the class, but it can be wrong!

so, SVM tries to create maximum seperation between data to get rid of the confusion of Probability. And the maximum seperation can be achieved through the distance between two closest vector lines from opp. sides.

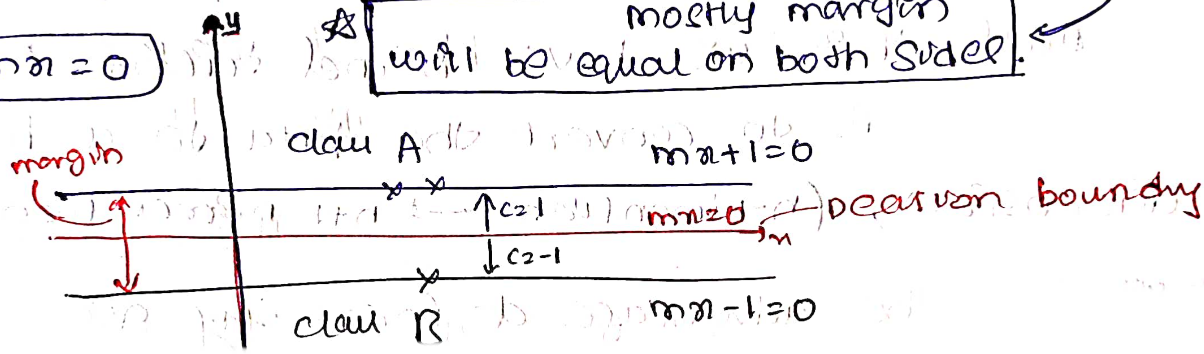



→ Decision boundary can be represented as, $y = mx + c$.

→ consider decision boundary is at origin, we get, and assume support vectors have $c = 1$.

$\boxed{mx = 0}$

* mostly margin will be equal on both sides.




→ Now, if we want to predict a new datapoint, then we just check the y value as is, if it falls in (0 to 1), it comes under class A.

or if it falls in (0 to -1), it comes under class B and we don't need probability, just by the range of y, we are predicting the new datapoint.

## ⊙ Soft margin and Hard Margin

→ the concept of SM & HM, refer to different approaches in handling the margin violations & the presence of noise or outliers in the data.

### Hard Margin

→ HM aims to find the optimal hyperplane that completely seperates the classes without any margin violations. It

→ It works well when the data is linearly seperable & free from noise.

$$\boxed{y_i(w^T x_i + b) \geq 1}$$

→ signifies all data points are correctly classified.

### Soft Margin

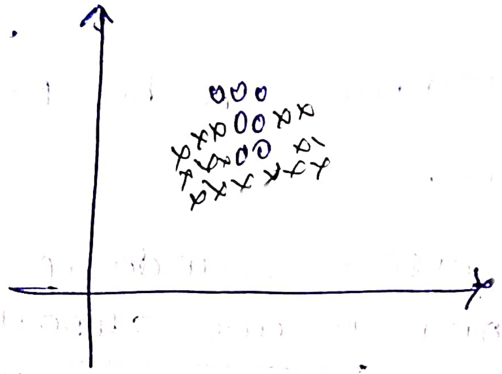
→ SM allows for some margin violations & misclassifications, thus providing flexibility when dealing with noisy/overlapping data. It introduces a slack variable ($\varepsilon_i$) to handle margin violations, allowing some points to fall within the margin / on wrong side of boundary.

$$\boxed{y_i(w^T x_i + b) \geq 1 - \varepsilon_i}$$

→ where, $\varepsilon_i \geq 0$ for all points.

## Kernelized support vector machines

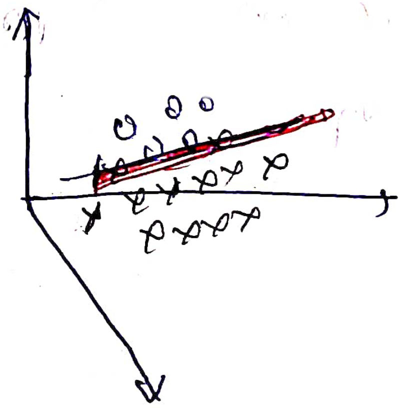
→ It's not always we have get which can be easily seperable by line!! usually, we will be having data in such a way that it cannot be seperable through lines easily! Such as below.



→ To deal this we have kernel tricks. Basically the idea is to convert the data to a higher dimension, (n-Dimension → n+1 Dimension).

→ What is the advantage of increasing dimension? If we take above example, there is no way to apply SVM, we can't get a decision boundary line as the points are clustered together & this is a 2D data.

Now, imagine it is converted to 3Dimension, then for sure we can get a layer or seperation between the data points.



⊙ How do we increase the dimension?

→ Say we have a 2D dataset as $(x, y)$. Now if we want to add another dimension, we can simply use some mathematical function to find $z$.

Eg $z = x + y$, $z = x - y$, $z = x^2 + y^2$, $z = x^2 - y^2$ etc.

→ So we can project the points in a 3D space and then derive a plane which divides the data into two parts. In theory, that's what a kernel function does without computing additional co-ordinates for the higher dimension.

## ⊙ Types of Kernels in SVM

→ There are many kernels, and here are a few popular ones.

1) Linear Kernel :- computes dot product between two feature vectors.

$$\boxed{K(x, y) = x^T y}$$

2) Polynomial Kernel :- $K(x, y) = (x^T y + c)^d$

3) Radial Basis Function (RBF) Kernel :-

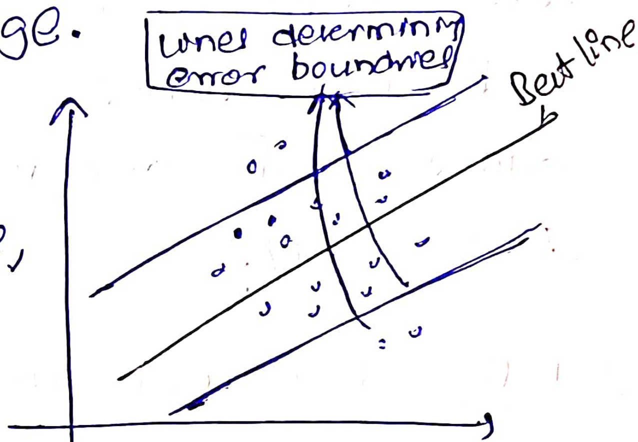
RBF kernel measures the similarity between two samples using a Gaussian radial function.

$$\boxed{K(x, y) = \exp(-\gamma \|x - y\|^2)}$$

# ⓪ Support vector Regressor

→ we know how linear Regression works, where we determine the best fit line. And in LR, the idea is to create a line which minimises the total Residual error.

→ In SVR, the approach is a bit different. Here, instead of trying to minimize the error, SVR focuses on keeping the error in a fixed range.




→ Here, the middle line is the best fit regressor line, and the other two lines are the bordering ones which denote the range of error.

→ The Best-fit line/hyperplane, will be the line which goes through the maximum number of data points and the error boundaries are chosen to ensure maximum inclusion.

→ So, we simply bring majority of data points in between the margin lines.