Lecture | DSML Advanced : SVM

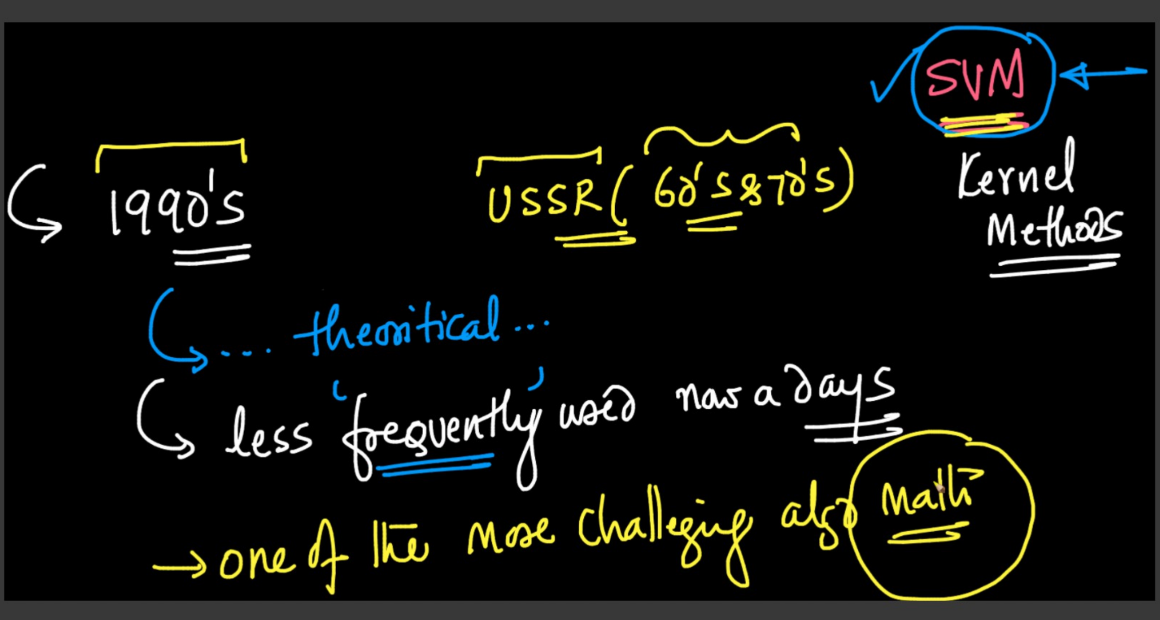
Lecture Link : https://www.scaler.com/meetings/i/dsml-advanced-svm-2/archive

Content

1. **SVM**
    - **Brief history** (1:39:15)
    - **General idea behind SVM**
    - **Geometric intuition** (1:42:20)
        - Linear SVM's
        - Hard Margin SVM's (2:06:00)
        - Soft Margin SVM's (2:18:00)

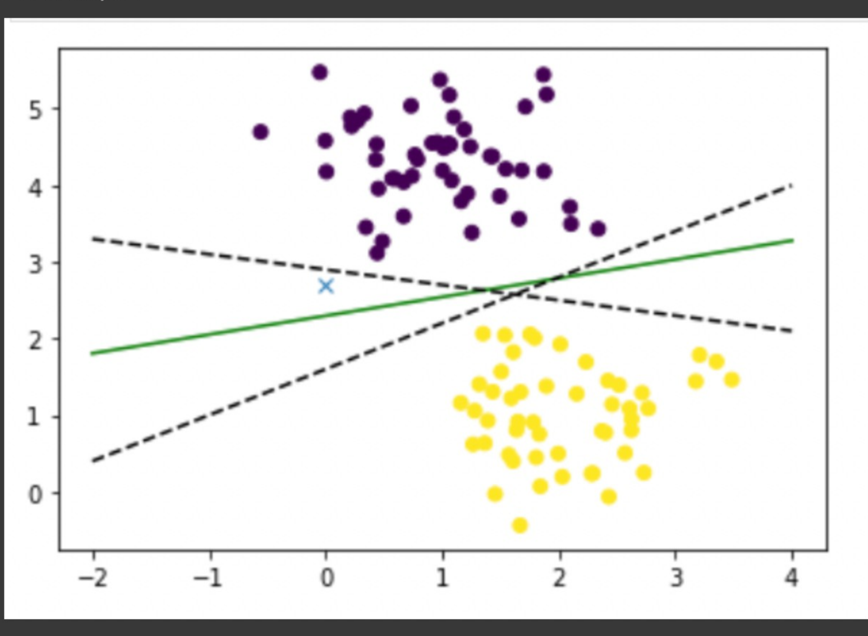
Brief History of SVM

- Inititally invented around 1960-70's by USSR but was brought to public notice during 1990's
- One of the most challenging algorithm in terms of mathematics
- Less frequently used nowadays because of more advanced models like boosting and bagging.



General idea behind SVM

In any classification problem, the main goal is to find a line/plane that can best seperate the two classes.
For example :



In the above graph you can see that depending on which line we choose, a test point (X) might be assigned a different label

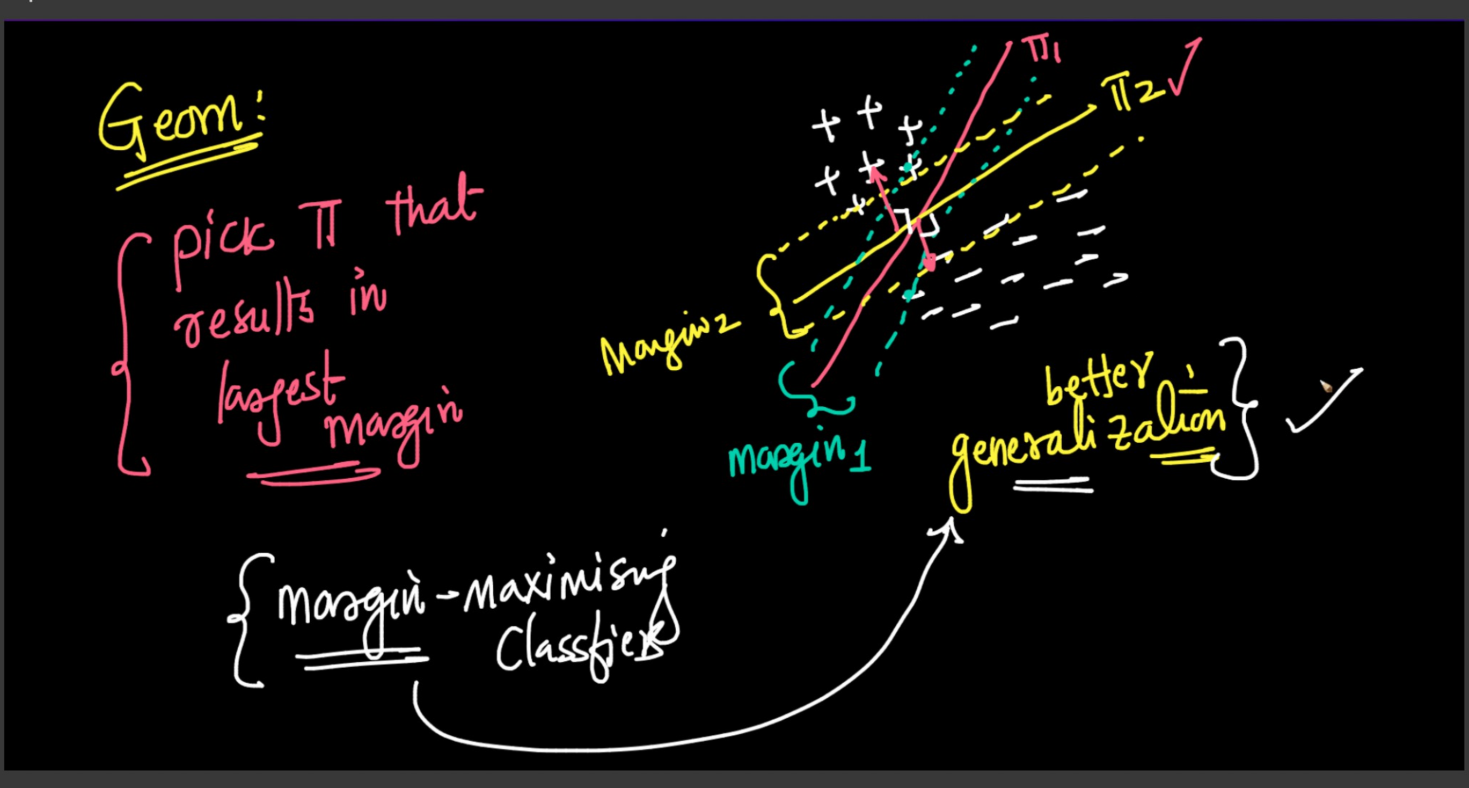
Observation:

1. There can be 1000 thousand such lines which can seperate these classes.
2. Drawing a line between classes is not enough, choosing a particular line may affect the performance during testing.

The main idea behind SVM algoithm is to find the line or plane that can best seperate the given classes.

Geometric intution behind SVM

For better understanding, let's consider a data set containg two classes as shown by positive(+) and negative(-) and the classes are linearly seperable.



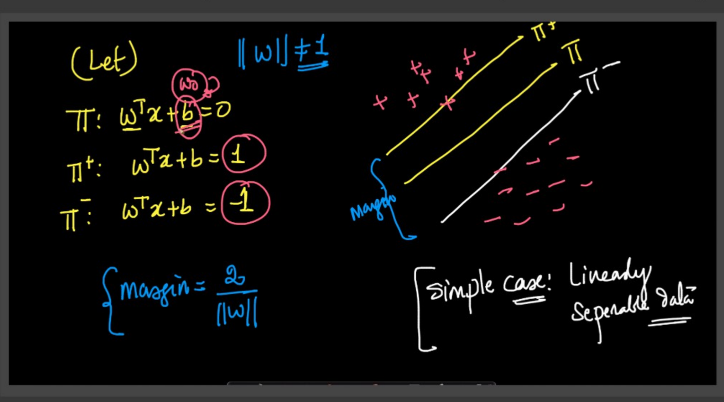
Our main goal is two find a line (**called as hyperplane**) with the largest margin possible, where margin can be defined as the distance between the hyperplane and the observations closest to the hyperplane (**called as support vectors**).

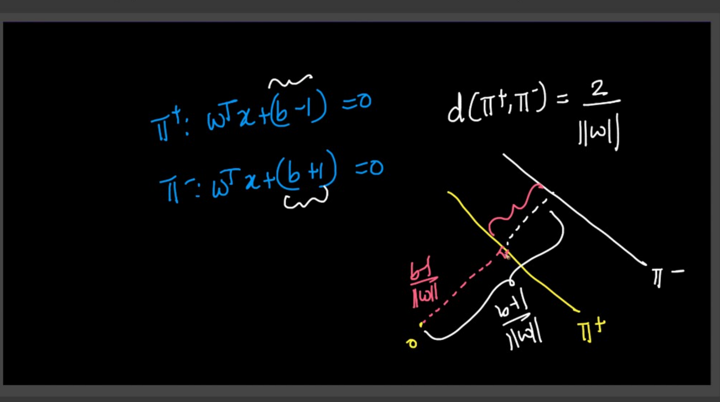
Now to understand maths behind it, let us consider the following variables:

> π : seperating hyperplane with maximum margin possible
> π+ : line passing through the positive data point closest to π
> π - : line passing through the negative data point closest to π

Here π+ and π- are called as support vectors.

All these can be interpreted in the form of some equations where margin can be simply calculated as distance between two parallel lines.



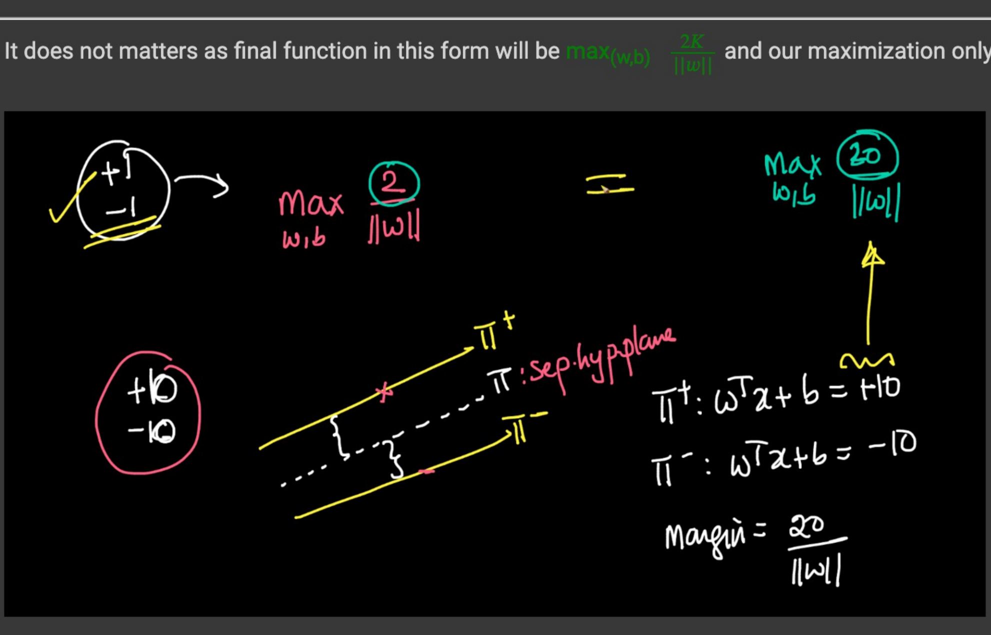


So the whole problem narrows down to maximizing this value such that all the data points are as far as possible from their respective support vectors :

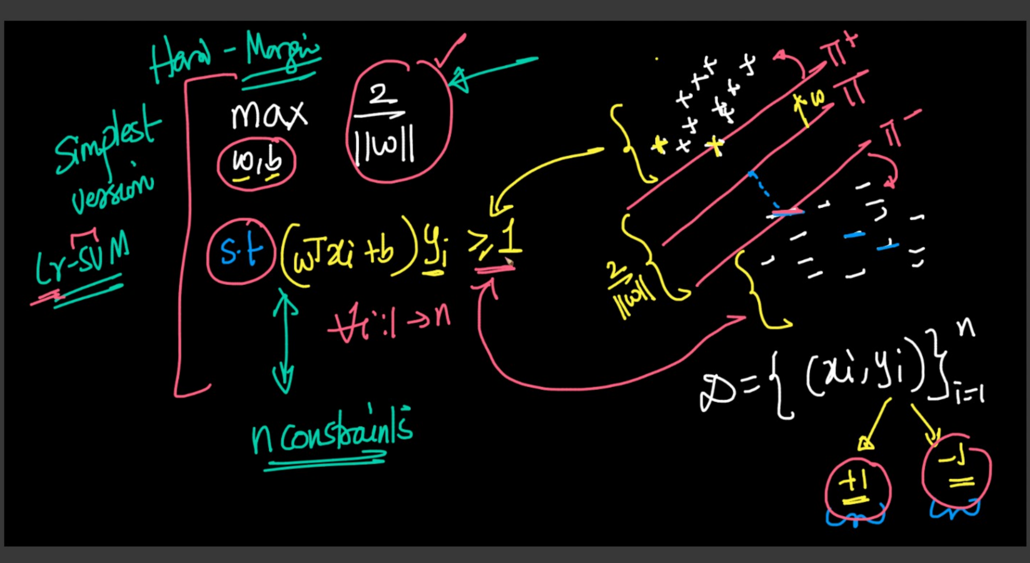
> $\max_{(w,b)} \frac{2}{||w||}$ or equivalent to $\min_{(w,b)} \frac{||w||}{2}$

Question: What if lines are of form Wx + b = K where k is some constant.

- It does not matters as final function in this form will be $\max_{(w,b)} \frac{2K}{||w||}$ and our maximization only depends on W .



- Above we studied are called as Linear - SVM's which is the simplest version of SVM with simple constraints
- SVM's are basically margin maximizing classifiers.
- This was the example of Hard Margin SVM's as our data is linearly seprable and SVM is very rigid in classification.



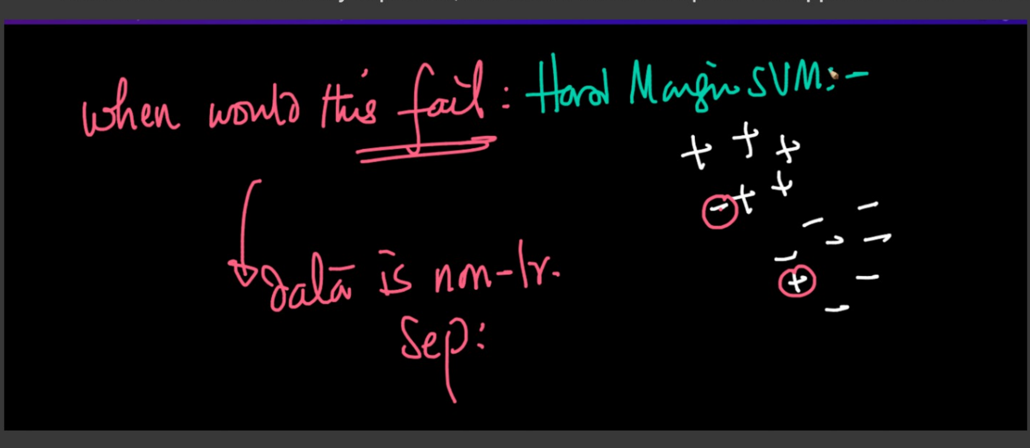
Linear SVM with a Hard Margin

If we strictly impose that all instances must be off the street (margin) and on the correct side, this is called hard margin classification. There are two main issues with hard margin classification:-

- First, it only works if the data is linearly separable.
- Second, it is sensitive to outliers.

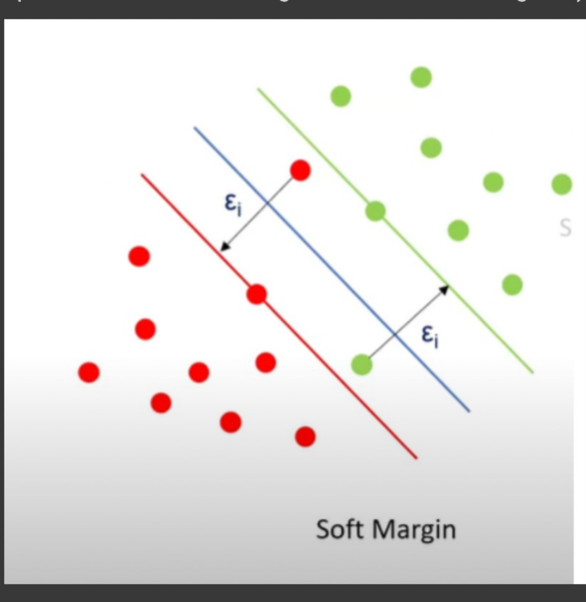
Question: When would Linear SVM with Hard Margin fails ?

- When data is almost linearly seperable, it contains some data points on opposite side of their correct space.



SVM with a Soft Margin

To avoid the issues of Hard Margin SVM, use a more flexible model. It is possible that data is almost linearly seperable which means few data points are either on the wrong side of margin or inside the margin.
The objective is to find a good balance between keeping the margin as large as possible and limiting the margin violations (i.e., points that end up in the middle of the margin or even on the wrong side). This is called soft margin classification.



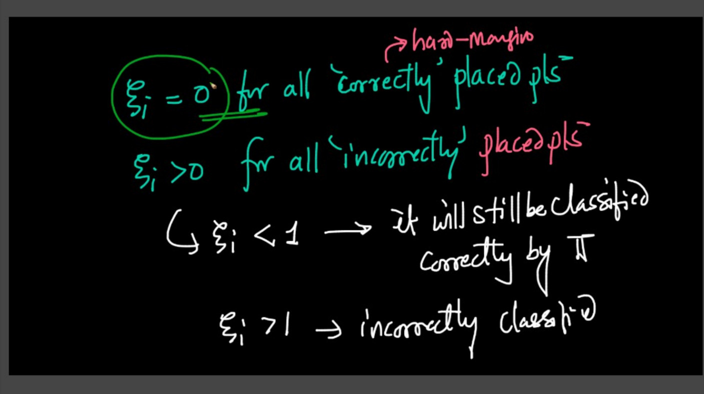
For this to understand, let's define ζ as error metric to penalize incorrectly placed data points.

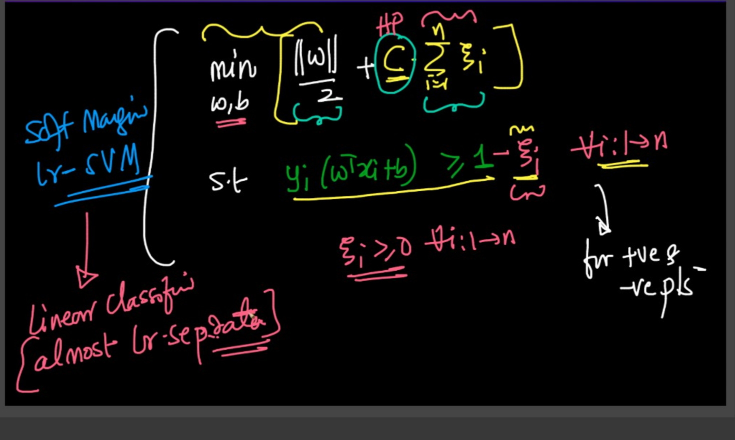
```
1) ζ = 0 for correctly placed data point
2) ζ > 0 and ζ < 1 which represents incorrectly placed data point but will be still be classified correctly.
3) ζ > 1 which represents incorrectly placed and  misclassified data point
```

Now with the introduction of ζ the final function will change to

> $\min_{(w,b)} \frac{||w||}{2} + \frac{C}{N} \sum_{i=1}^{N} \zeta_i$

Where C is hyperparamter, and we need to minimize mean of all incorrectly place points error to get best classification.





Above equation holds true for both positive as well as negative points which is explained in calculations below: