Here w is a vector perpendicular to pie, note w is not a unit vector.

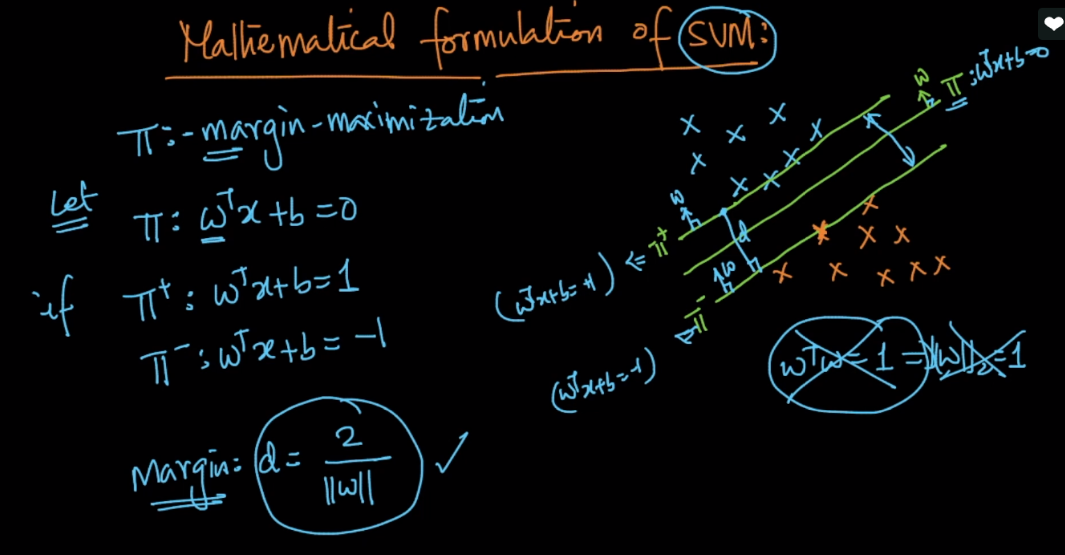
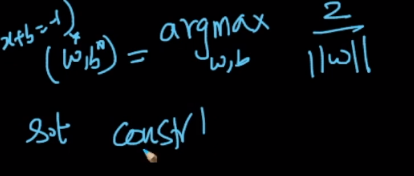
The equation of plane pie (W\_T \* x\_i) + b = 0

Now if equation of pie+ is (W\_T \* x\_i) + b = 1, that means pie+ is at distance 1 from pie.

And if equation of pie- is (W\_T \* x\_i) + b = -1, that means pie- is at distance 1 from pie in opposite direction of pie.

The distance between pie+ and pie- or margin distance will be 2/||W||.

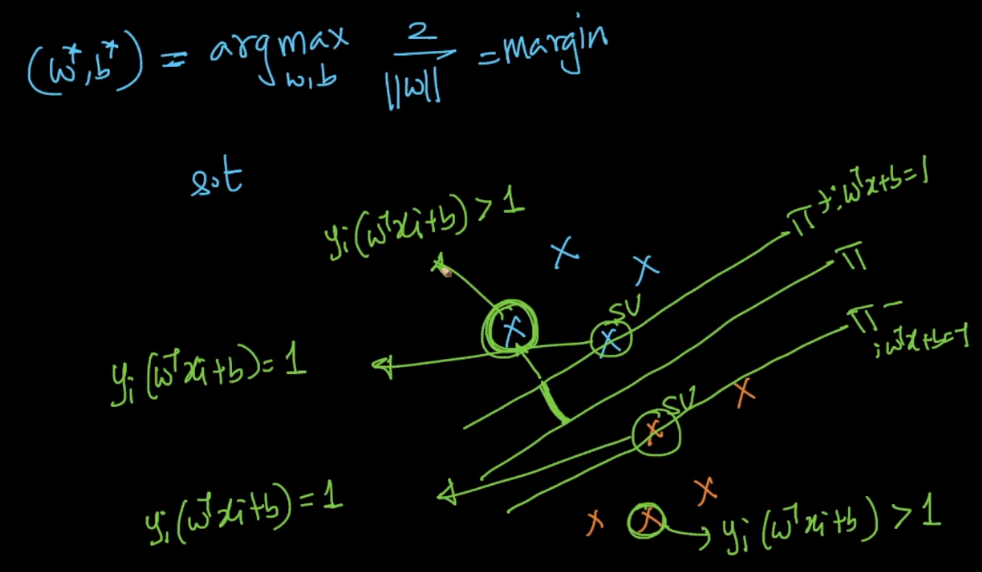
Now our objective function is to find w where 2/||W|| is maximum, because we want margin dist to be maximum. Along with constraints, that all +Ve points are on or above pie+ and all –ve points are below or on pie-

Given any point if it’s in positive side then y\_i \* ((W\_T \* x\_i) + b) >= 1, because y\_i = +Ve, and (W\_T \* x\_i) + b >= 1

And if it’s in negative side then y\_i \* ((W\_T \* x\_i) + b) >= 1, because y\_i = -Ve, and (W\_T \* x\_i) + b <= -1.

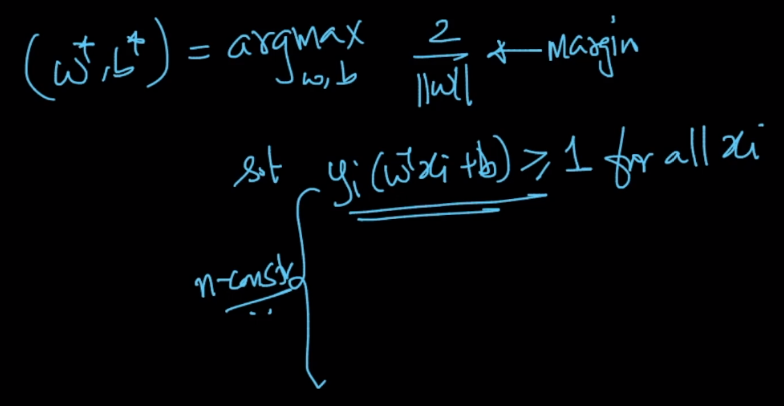
Note we are finding distance of any point from pie and not from pie+ or pie-.



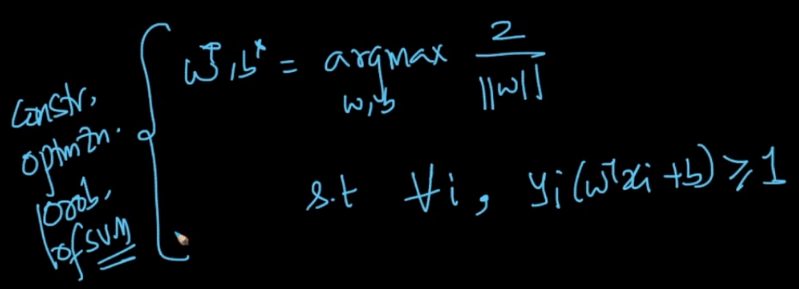
And therefore we can say that our constraint which is all the points, lie in their region, that means +Ve points lie above or on pie+, and –ve point lie below or on pie-, mathematically as.

y\_i \* ((W\_T \* x\_i) + b) >= 1.

This is not a single constraints, these are ‘n’ constraints one for each point.



And therefore since our objective function has constraints, it becomes constraint optimization problem of SVM.

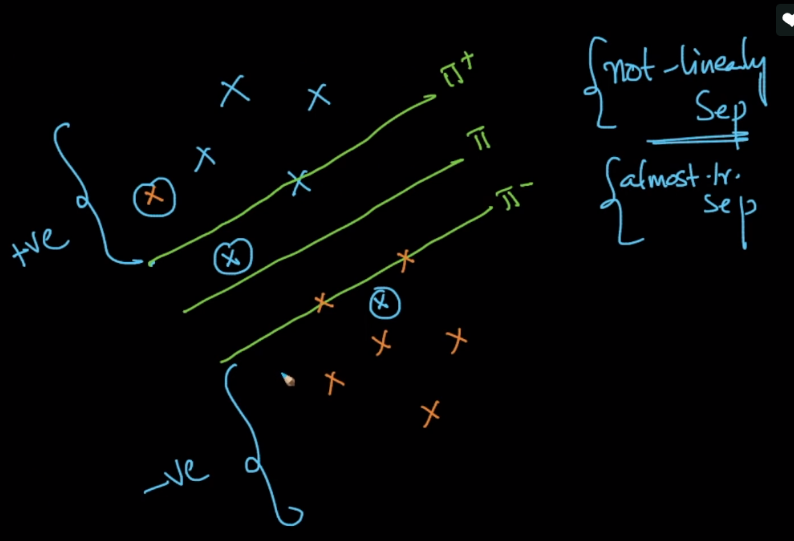


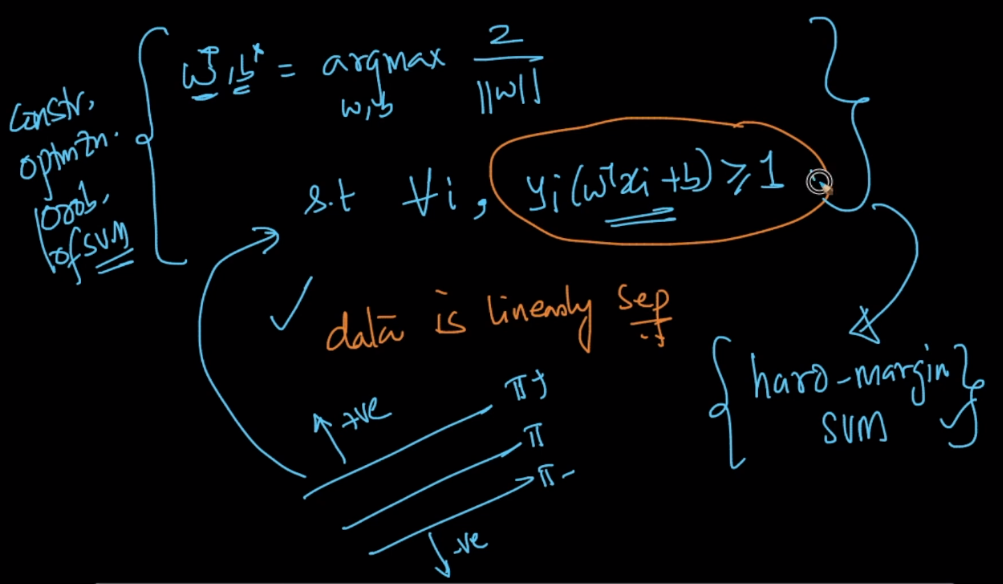
But we will not able to find w and b where margin dist is max if above constraint is not completely satisfied, or when data is not-linearly seperable or almost linearly seperable. See data points given in below image.

Here one –ve point lie in +Ve region, one +Ve lie b/w pie and pie+, one +Ve point lie in negative region.

So for such data we would not able to find w and b.

And therefore objective function is called **Hard margin SVM,** because here we are enforcing that datapoints should satisfy above constraint.





Now how we apply SVM for such dataset, we’ll use **soft margin SVM. How we do it?**

* Let’s take the +Ve point lie b/w pie and pie+.

It’s distance from pie is 0.5 ie **y\_i \* ((W\_T \* x\_i) + b) = 0.5**

We can write it as: **y\_i \* ((W\_T \* x\_i) + b) = 1 – (0.5)**

* Let’s take the +Ve point lie b/w pie and pie-

It’s distance from pie is 0.5 ie **y\_i \* ((W\_T \* x\_i) + b) = -0.5**

We can write it as **y\_i \* ((W\_T \* x\_i) + b) = 1 – (1.5)**

* Let’s take the +Ve point lie below pie-

It’s distance from pie is 1.5 ie **y\_i \* ((W\_T \* x\_i) + b) = -1.5**

We can write it as **y\_i \* ((W\_T \* x\_i) + b) = 1 – (2.5)**

* Let’s take the -Ve point lie above pie+

It’s distance from pie is 1.5 ie **y\_i \* ((W\_T \* x\_i) + b) = -1.5**

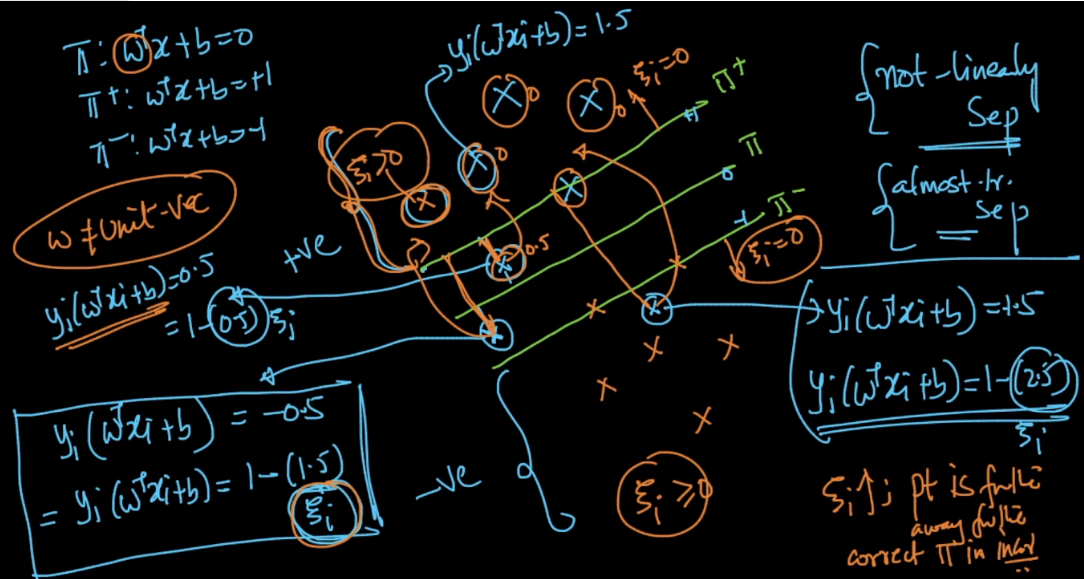
We can write it as **y\_i \* ((W\_T \* x\_i) + b) = 1 – (2.5)**

Now let’s take zeta as 1 – (zeta), in all above 4 terms. Here zeta is the distance of the point from the plane of the class, in which class that point belong. Example let’s take for 4th point, here zeta is 2.5 that means it’s 2.5 distance away from pie-.

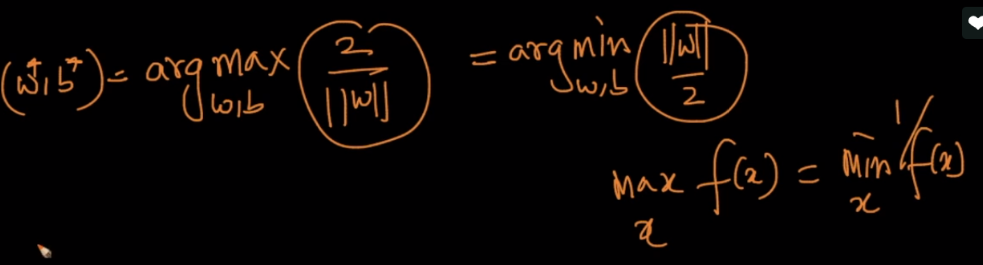
Similarly for 1st point, zeta is 0.5 since it’s 0.5 away from pie+.

For 2nd point, zeta is 1.5, and for 3rd point, zeta is 2.5

So here zeta is basically showing the the distance of wrongly classified points, and we want sum of all zeta to be as minimum as possible.



Note: argmax 2 / ||W|| = armin ||W|| / 2.



Now our new objective function will be as given in below image.

Here left part is 1/margin and right part is average of zeta along with **c** which is hyperparameter.

So what we are doing in this objective function is:

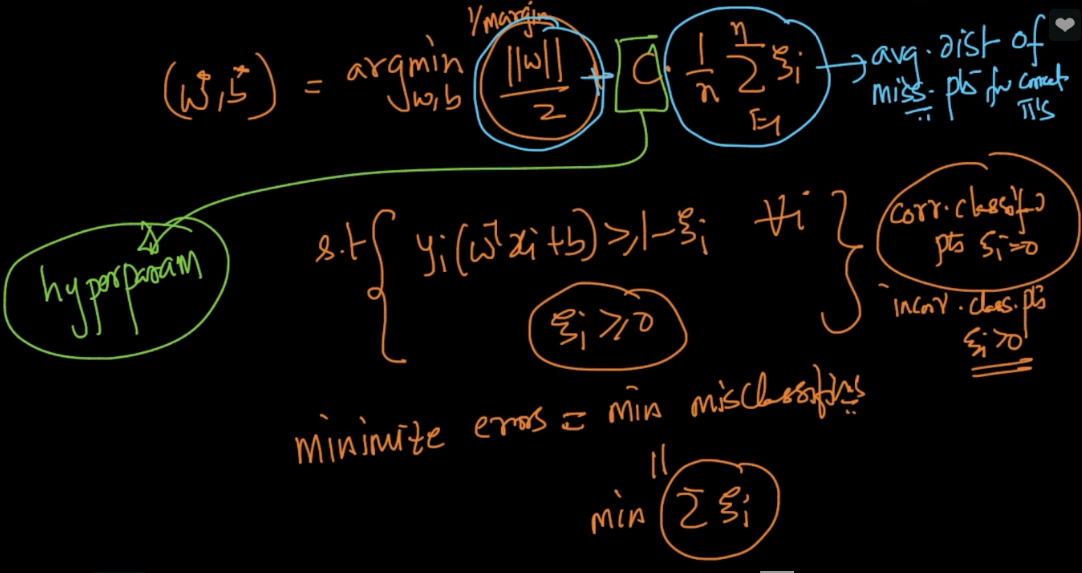
We are trying to minimize sum of left and right part where:

Minimizing left part, that mean minimizing 1/margin, that mean maximizing margin distance.

Minimizing right part, that means minimizing zeta or minimizing wrongly classified points.

**Now what would be zeta value for accurate classified points.**

For such classified points, zeta will be 0, therefore considering this constraint for soft margin SVM wil be **y\_i \* ((W\_T \* x\_i) + b) >= 1 – zeta and zeta >= 0**

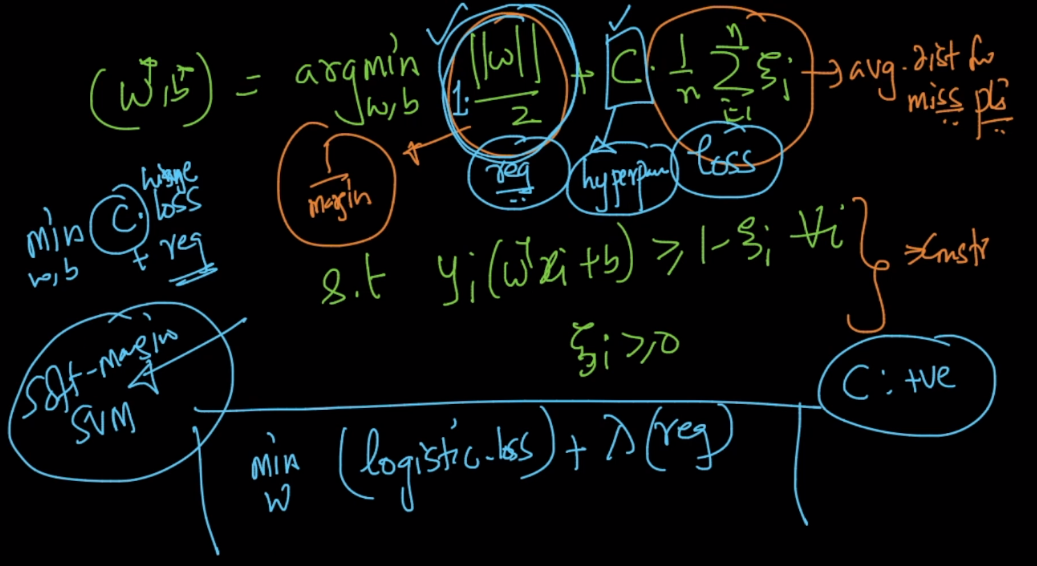


Now this new objective function can be seen as

Right part of averaging of zeta is **loss function**, since zeta is the loss/error we are having in model.

C is hyperparameter.

Left part is regularization term.

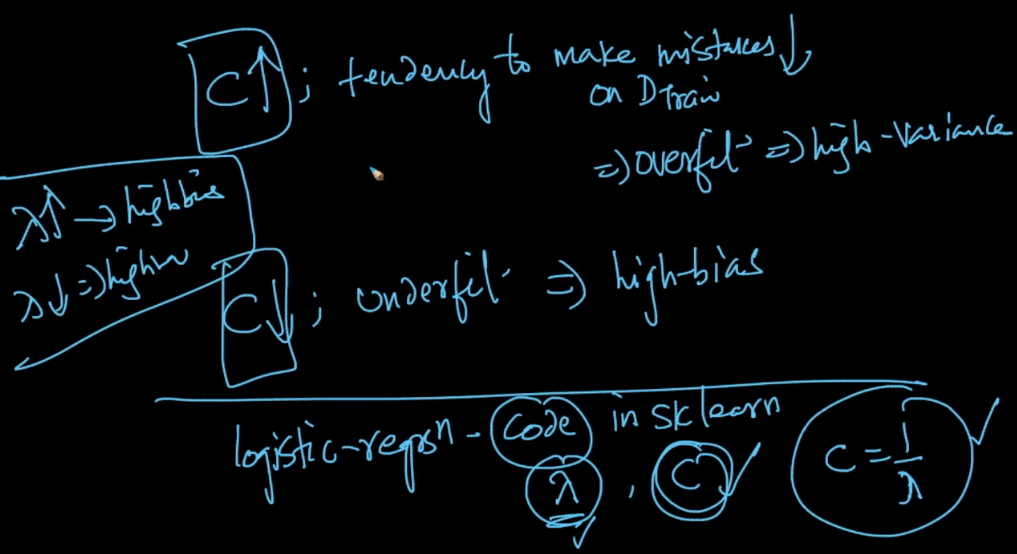


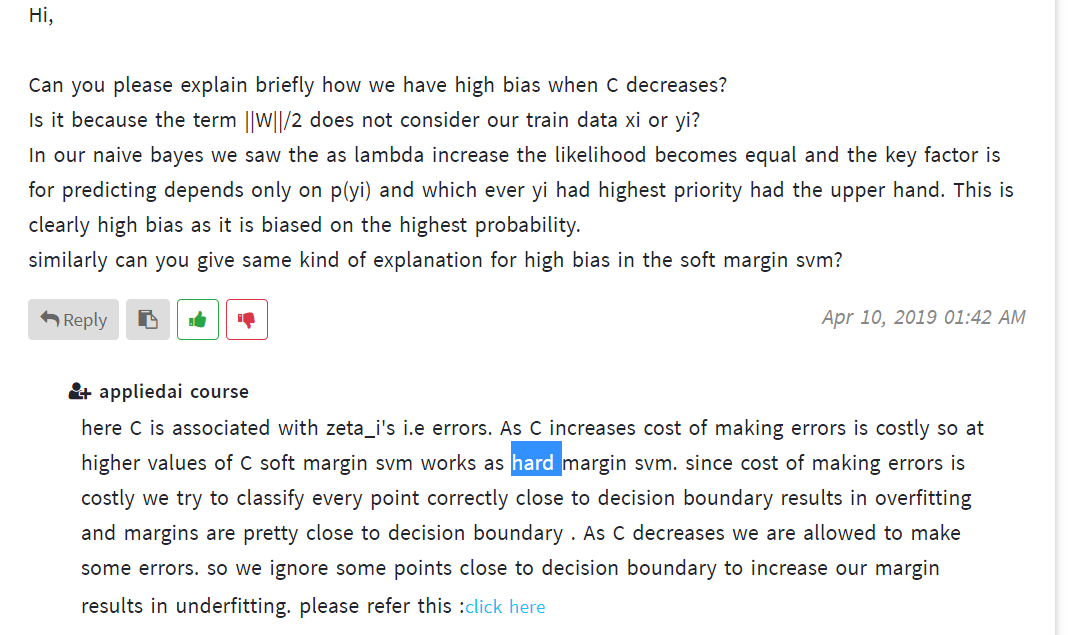
Here C will play a role of maintaining a balance between overfit and underfit.

If C increases that means we are giving more importance to loss function, and our objective function will only try to minimize loss function in this case, therefore it would be overfit. As we are making almost zero error for all points, even for wrong classified points.

And if C decreases we’ll only focus on minimizing 1/margin, that means increasing margin, and in doing this it would do underfit, as there will be lot of points, which we’ll be minclassified now.

Here C in SVM is just opposite to lambda in logistic regression.





**How dist b/w pie+ and pie- is calculated:**

