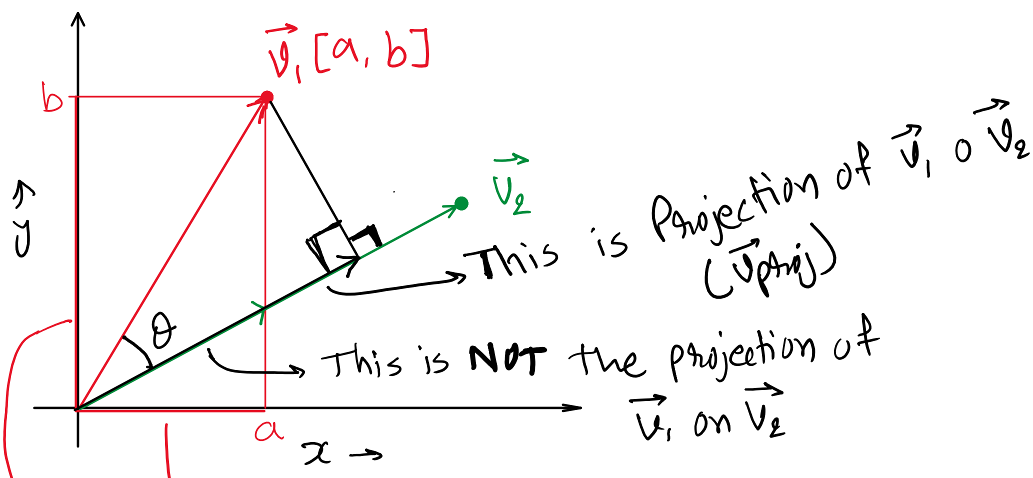
Linear Algebra 3

21 March 2025 10:18

☆ Projection of a vector



x-component of $\vec{v}_1$ $\parallel$ Projection of $\vec{v}_1$ on x-axis

y-component of $\vec{v}_1$ = Projection of $\vec{v}_1$ on Y-axis

Let θ be the angle subtended by $\vec{v}_1$ & $\vec{v}_2$ and, $\vec{v}_{proj}$ is the projection vector of $\vec{v}_1$ on $\vec{v}_2$

$$\therefore \cos\theta = \frac{\|\vec{v}_{proj}\|}{\|\vec{v}_1\|}$$

$$\boxed{\therefore \|\vec{v}_{proj}\| = \|\vec{v}_1\| \cdot \cos\theta}$$

Magnitude of projection vector

For any vector $\vec{a}$, $\hat{a} = \dfrac{\vec{a}}{\|\vec{a}\|} \Rightarrow \vec{a} = \|\vec{a}\| \cdot \hat{a}$

where $\hat{a}$ is the unit vector in the direction of $\vec{a}$.

$\therefore \vec{v}_{proj} = \|\vec{v}_{proj}\| \cdot \hat{v}_{proj}$ where $\hat{v}_{proj}$ = unit vector in the direction of $\vec{v}_{proj}$

But $\hat{v}_{proj} = \hat{v}_2$ because $\vec{v}_{proj}$ is in the same direction of $\vec{v}_2$

$\therefore \vec{v}_{proj} = \|\vec{v}_{proj}\| \cdot \hat{v}_2$ where $\hat{v}_2 = \dfrac{\vec{v}_2}{\|\vec{v}_2\|}$ & $\|\vec{v}_{proj}\| = \|\vec{v}_1\| \cdot \cos\theta$

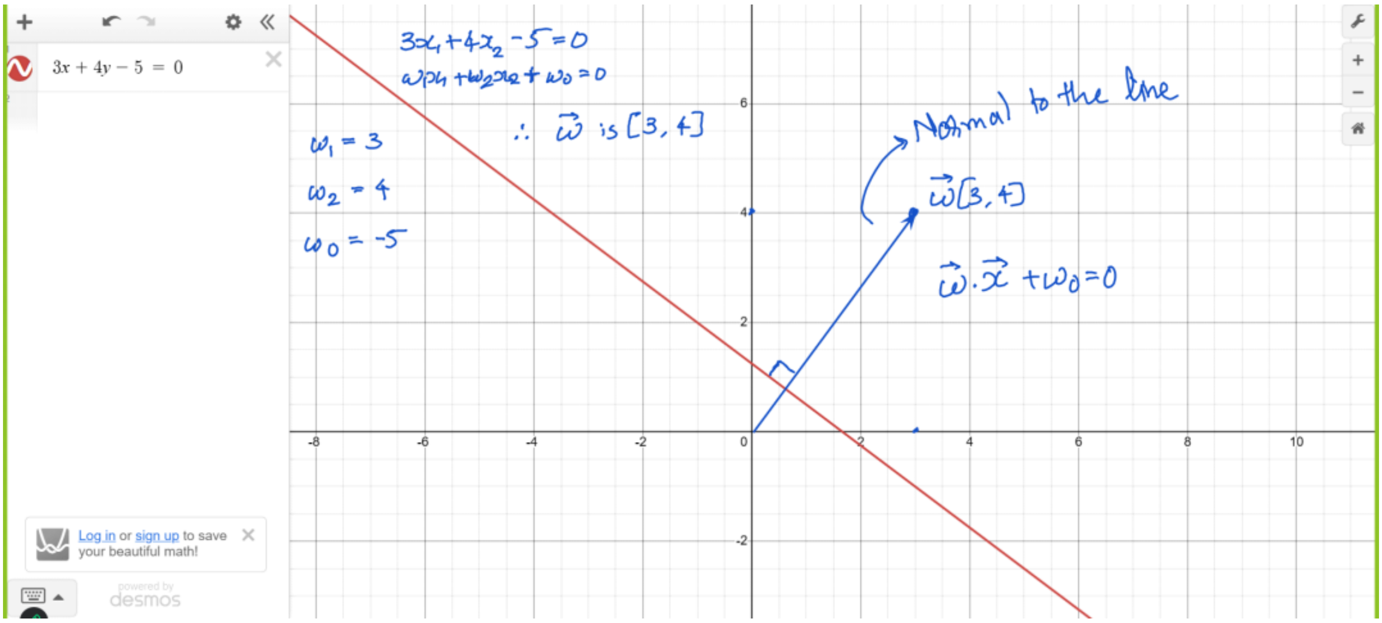
$$\therefore \quad \overrightarrow{V_{proj}} = \|\vec{v_1}\| \cos\theta \cdot \frac{\vec{V_2}}{\|\vec{V_2}\|}$$

Normal Equation of a Line:

$ax + by + c = 0$ OR $w_1 x_1 + w_2 x_2 + \dots + w_n x_n + w_0 = 0$

$\vec{w} \cdot \vec{x} + w_0 = 0$

where $\vec{w}$ is $[w_1, w_2, w_3 \dots, w_n]$



Let w [w1, w2, w3, ... , wn] & x [x1, x2, x3, ... , xn] then how can we multiply them (recall vector/matrix multiplication)?

$[2 \quad 4 \quad 5] \times [1 \quad 2 \quad 6]$ are not compatible for matrix multiplication. ∴ we take transpose of one of them.

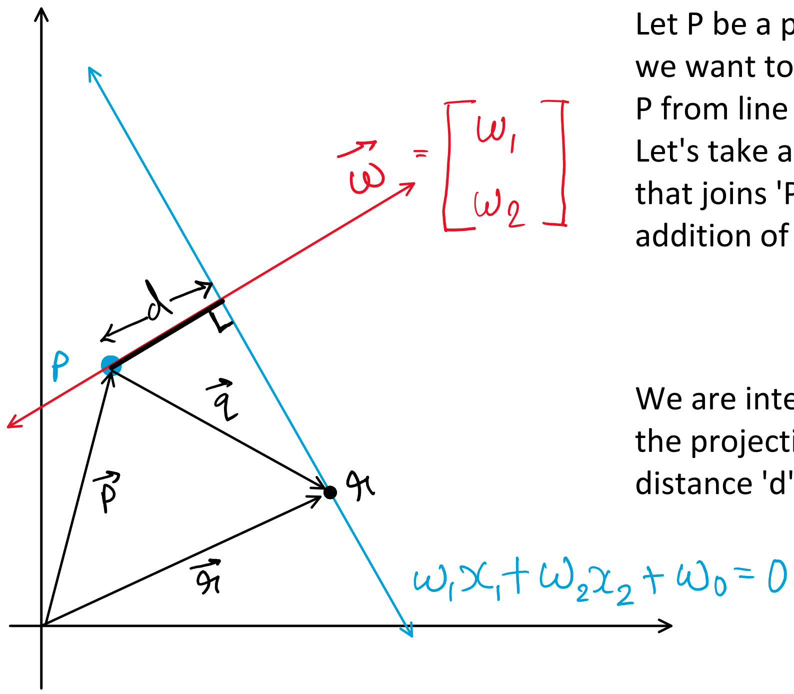
$$\begin{bmatrix} 2 \\ 4 \\ 5 \end{bmatrix} \times [1 \quad 2 \quad 6] = (2\times1) + (4\times2) + (5\times6)$$

Similarly, $[w_1 \quad w_2 \quad w_3]$ & $[x_1 \quad x_2 \quad x_3]$ are not compatible for matrix multiplication but $\vec{w}^T$ & $\vec{x}$ are compatible

$$\therefore \vec{w}^T \cdot \vec{x} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \cdot [x_1 \; x_2 \; x_3] = w_1 x_1 + w_2 x_2 + w_3 x_3$$

$\therefore$ The Normal Eqⁿ will be $\boxed{\vec{w}^T \cdot \vec{x} + w_0 = 0}$

☆ Distance of a point from the line



Let P be a point & l=w1x1 + w2x2 + w0 = 0 be the line and we want to find the distance (perpendicular distance) of P from line l.

Let's take any point 'r' on the line, if we create a vector that joins 'P' to 'r' & call it $\vec{q}$ then we can write (as per addition of two vectors):

$$\vec{p} + \vec{q} = \vec{r}$$

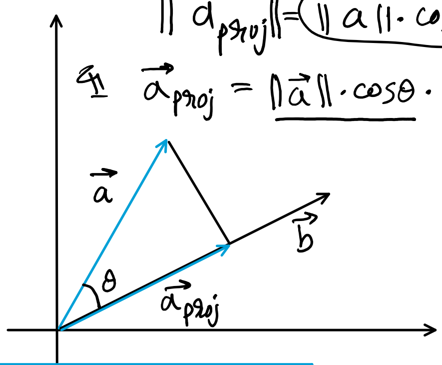
We are interested in finding $\vec{q}$ because magnitude of the projection vector of $\vec{q}$ on $\vec{w}$ is nothing but the distance 'd' of the point from the line 'l'.

$$\therefore \vec{q} = \vec{r} - \vec{p}$$

An Interesting result about projection vector:

$$\|\vec{a}_{proj}\| = \|\vec{a}\| \cdot \cos\theta$$

$$\vec{a}_{proj} = \underline{\|\vec{a}\| \cdot \cos\theta} \cdot \hat{b}$$



$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \cdot \|\vec{b}\| \cos\theta$$

$$\therefore \|\vec{a}\| \cdot \cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{b}\|}$$

$$\therefore \vec{a}_{proj} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{b}\|} \cdot \hat{b}$$

AND $\boxed{\|\vec{a}_{proj}\| = \dfrac{\vec{a} \cdot \vec{b}}{\|\vec{b}\|}}$

$$\vec{a}_{proj} = \frac{\vec{a} \cdot \vec{b}}{\|\vec{b}\|} \cdot \frac{\vec{b}}{\|\vec{b}\|}$$

$$\boxed{\therefore \|\vec{d}\| = \|\vec{q}_{proj}\| = \frac{\vec{q}\cdot\vec{w}}{\|\vec{w}\|}} \quad \text{I}$$

The only problem with this equation is that we do not know anything about $\vec{q}$ then how will we find distance. Therefore, we need to convert this equation into a form where it has quantities that we know e.g., coordinates of point 'p'.

Let p(x1, y1) & r(x2, y2) then p = x1$\cdot\hat{i}$ + y1$\cdot\hat{j}$ & q = x2$\cdot\hat{i}$ + y2$\cdot\hat{j}$

And, $\vec{q} = \vec{r} - \vec{p} \Rightarrow \vec{q} = x_2\cdot\hat{i} + y_2\cdot\hat{j} - x_1\hat{i} - y_1\cdot\hat{j}$

$$\boxed{\vec{q} = (x_2 - x_1)\cdot\hat{i} + (y_2 - y_1)\cdot\hat{j}} \quad \text{A}$$

Similarly, $\boxed{\vec{w} = w_1\cdot\hat{i} + w_2\cdot\hat{j}}$ B & $\boxed{\|\vec{w}\| = \sqrt{w_1^2 + w_2^2}}$ C

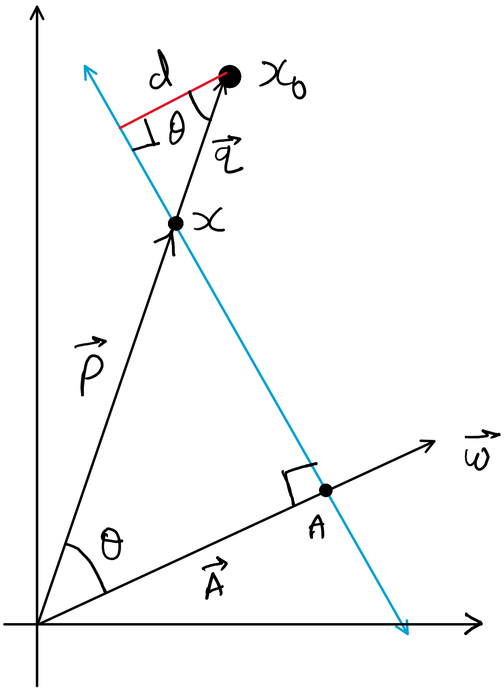
Substituting A, B & C into I:

$$\|\vec{d}\| = \frac{\left[(x_2 - x_1)\cdot\hat{i} + (y_2 - y_1)\cdot\hat{j}\right]\cdot\left[w_1\cdot\hat{i} + w_2\cdot\hat{j}\right]}{\sqrt{w_1^2 + w_2^2}}$$

From the dot product of two vectors $\vec{x}(a_1, b_1)$ & $\vec{y}(a_2, b_2) \Rightarrow \boxed{\vec{x}\cdot\vec{y} = a_1\cdot a_2 + b_1\cdot b_2}$

$$\boxed{\therefore \|\vec{d}\| = \frac{(x_2 - x_1)\cdot w_1 + (y_2 - y_1)\cdot w_2}{\sqrt{w_1^2 + w_2^2}}}$$

Another formula for distance of a point from the line:



$$\|\vec{x_0}\| = \|\vec{p}\| + \|\vec{q}\| \Rightarrow \boxed{\|\vec{q}\| = \|\vec{x_0}\| - \|\vec{p}\|} \quad \text{— I}$$

Considering the lower triangle:

$$\cos\theta = \frac{A}{\|\vec{p}\|}$$

$$\boxed{\therefore \|\vec{p}\| = \frac{A}{\cos\theta}} \quad \text{— II}$$

Considering the upper triangle:

$$\cos\theta = \frac{d}{\|\vec{q}\|} \Rightarrow \boxed{d = \|\vec{q}\|\cos\theta} \quad \text{— III}$$

$$d = (\|\vec{x_0}\| - \|\vec{p}\|) \cdot \cos\theta \qquad (\text{From I})$$

$$\therefore d = \left(\|\vec{x_0}\| - \frac{A}{\cos\theta}\right) \cdot \cos\theta \qquad (\text{From II})$$

$$\boxed{\therefore d = \|\vec{x_0}\| \cdot \cos\theta - A} \quad \text{— IV}$$

Now, $\vec{\omega}^T \cdot \vec{x}_0 = \|\vec{\omega}^T\| \cdot \|\vec{x_0}\| \cdot \cos\theta$

$$\therefore \|\vec{x}_0\| = \frac{\vec{\omega}^T \cdot \vec{x_0}}{\|\vec{\omega}^T\| \cdot \cos\theta} \qquad \text{OR} \qquad \boxed{\cos\theta = \frac{\vec{\omega}^T \cdot \vec{x_0}}{\|\vec{\omega}^T\| \cdot \|\vec{x}_0\|}} \quad \text{— V}$$

Putting this value of $\cos\theta$ into IV

$$\boxed{d = \frac{\cancel{\|\vec{x_0}\|} \cdot \vec{\omega}^T \cdot \vec{x_0}}{\|\vec{\omega}^T\| \cdot \cancel{\|\vec{x_0}\|}} \quad \text{— A}} \quad \text{VI}$$

In the lower triangle, $\|\vec{A}\| = \|\vec{P}\| \cdot \cos\theta$ OR

$$\|\vec{A}\| = \|\vec{x}\| \cdot \cos\theta$$

Putting value of $\cos\theta$ from V:

$$\|\vec{A}\| = \|\vec{x}\| \cdot \frac{\vec{\omega}^T \cdot \vec{x_0}}{\|\vec{\omega}^T\| \cdot \|\vec{x_0}\|}$$

Replacing $\vec{x_0}$ by $\vec{x}$ as both of them subtend the same θ on $\vec{\omega}$

$$\therefore \|\vec{A}\| = \frac{\cancel{\|\vec{x}\|} \cdot \vec{\omega}^T \cdot \vec{x}}{\|\vec{\omega}^T\| \cdot \cancel{\|\vec{x}\|}}$$

But equation of our Hyper plane is: $\omega_1 x_1 + \omega_2 x_2 + \omega_0 = 0$

OR $\vec{\omega}^T \cdot \vec{x} + \omega_0 = 0 \Rightarrow \vec{\omega}^T \cdot \vec{x} = -\omega_0$

$$\boxed{\therefore \|\vec{A}\| = \frac{-\omega_0}{\|\vec{\omega}\|}} \quad \text{VII}$$

Putting this value into VI:

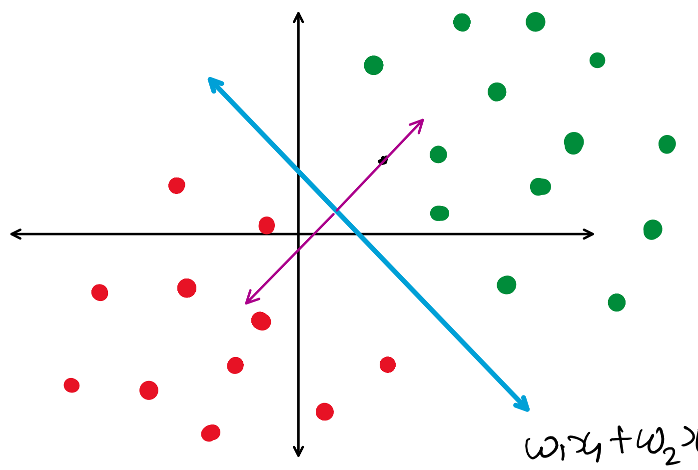
$$d = \frac{\vec{\omega}^T \cdot \vec{x}}{\|\vec{\omega}\|} - \frac{(-\omega_0)}{\|\vec{\omega}\|}$$

$$\boxed{d = \frac{\vec{\omega}^T \cdot \vec{x} + \omega_0}{\|\vec{\omega}\|}}$$ V.I.M.P. Result.

Significance of the direction of $\vec{\omega}$:

● → +ve class

● → -ve class



Let's take two lines:

(i) $x_1 + x_2 + 3 = 0$

$\therefore \omega_1 = 1, \omega_2 = 1, \omega_0 = 3$

$\therefore \vec{\omega} = [1, 1]$

(ii) $-x_1 - x_2 - 3 = 0$

$\therefore \omega_1 = -1, \omega_2 = -1, \omega_0 = -3$

$\therefore \vec{\omega} = [-1, -1]$

But both of these lines are exactly the same and their $\vec{\omega}$ are opposit to each other.

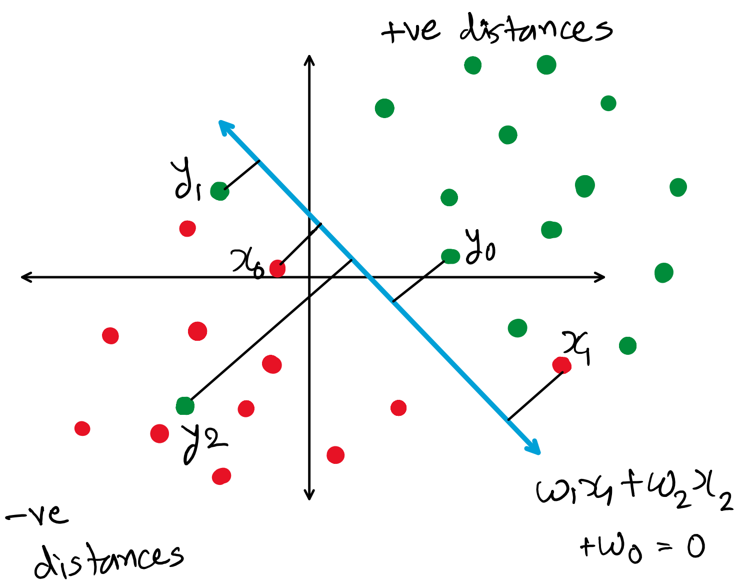
The direction of $\vec{\omega}$ shows the area of +ve class points.

Loss Minimization

As we know, the line (separator) that is more distant from the data points is a better hyperplane. So the goal is to **maximize** the distances of the datapoints from our line. Because more the distances, less is the chance of future error.

The distance of one point from the line is: $\frac{\vec{w}^T \cdot \vec{x} + w_0}{\|\vec{w}\|}$

Therefore, total distance (distances of all the points from the line) can be given as:



$$\sum_{i=1}^{n} \frac{\vec{w}^T \vec{x}_i + w_0}{\|\vec{w}\|}$$

And we want to maximize this distance. But logic has got two problems:

1. +ve and -ve distances will cancel out each other
2. Misclassified points are not penalized instead, they are appreciated.

To solve the first issue, we can either take absolute values of distances (modulus) or we may also square the distances and that will convert all the -ve distances to +ve but both of these ways will only encourage the distances of misclassified points.

An easy & effective solution: multiply the distance with its actual class.
Example:
Correctly classified points: x0 & y0
Suppose distance of point x0 from the line is -2 units. Actual label of x0 is -ve (-1) and we have also classified it as -ve class.
The distance of x0 = -2
The actual label of x0 = -1
Now multiply them: (-2) * (-1) = +2
(solution of problem 1, distances of all correctly classified red points will become positive)

Suppose distance of y0 = +3
Actual class of y0 = +1
Multiplication: (+3) * (+1) = +3
This means, the distances of all correctly classified green points will remain positive)

Misclassified points: x1, y1 & y2
Actual label of x1 = -1
But the distance of x1 = +2
Multiplication: (+2) * (-1) = -2
This means, this logic is penalizing the misclassified red points!

Actual label of y1 = +1
But the distance of y1 = -1.5
Multiplication: (+1) * (-1.5) = -1.5
This means, this logic is also penalizing the misclassified green points!

Actual label of y2 = +1
But the distance of y2 = -5
Multiplication: (-5) * (+1) = -5
It means our logic is penalizing distant misclassified points more than the misclassified points which are closer to the line.

features (x_i) label (y_i)

	f_1	f_2	f_3 - - - -	f_d	y	
x_0						y_0
x_1						y_1
x_2						y_2
⋮						⋮
x_n						y_n

Implementing this logic in the previous formula:

$$\sum_{i=1}^{n} \frac{\vec{w}^T \cdot \vec{x}_i + w_0}{\|\vec{w}\|} \cdot y_i$$

We call this function "**Gain Function**" as we seek to maximize the value of this function.

$$\boxed{\sigma(\vec{x}, \vec{w}, \vec{y}, w_0) = \sum_{i=1}^{n} \frac{\vec{w}^T \cdot \vec{x}_i + w_0}{\|\vec{w}\|} \cdot y_i}$$

But we wanted to do "**Loss Minimization"** & not the **"Gain Maximization"**! Hence, we will convert this Gain Function into **"Loss Function"** just by adding a -ve sign before it.

$$\boxed{L(\vec{w}, w_0) = -\sum_{i=1}^{n} \frac{\vec{w}^T \cdot \vec{x}_i + w_0}{\|\vec{w}\|} \cdot y_i}$$

We did not write L as a function of $\vec{x}_i$ & $\vec{y}_i$ because $\vec{x}_i$ are the features & $\vec{y}_i$ are the labels which will be provided to us in order to calculate loss of that line.

Now, how can we minimize this loss?
Meaning of loss minimization is that we want to find a line for which this loss function has smallest value.

So the question is: How can we identify that line?
There are two ways to get the answer this question:

1. Using brute force
2. Using some other "intelligent" technique

Using Brute Force:
In this way, we will calculate value of Loss Function for each possible values of w & w0 and finally the values of w & w0 corresponding to the minimum value of Loss Function will be our choice.
Let's imagine that there are 5 features (hence 5 elements in w) and the possible values of each w_i is in the interval [1, 10] (including both) with the step of 1. Also for w_0 we are keeping this assumption. Then how many different lines can we get using these values of w & w_0? Because we will need to calculate loss for each of these lines.

w1	1	2	3	4	5	6	7	8	9	10
w2	1	2	3	4	5	6	8	8	9	10
w3										
w4										
w5										
w0	1	2	3	4	5	6	7	8	9	10

This way we will end up getting 10^6 different lines and it will not be a small task to compute loss of each one of them & to find the minimum loss line.

But imagine what if the possible values of each w_i is increasing with step size of 0.01?

Now we will get 1000 different possible values of each w_i making total number of lines 1000^6 which will be very much time consuming.
Now imagine instead of 5 features, we have 50 features (plus one w_0) each can have values in the interval [-1000, 1000] with the step size of 0.001 then we will have $20,00,000^{51}$ lines. To evaluate loss for each of these lines & to find the minimum loss line will be next to impossible even for modern computers.
Therefore, using brute force is not the best idea.
Hence we need a more intelligent way of doing it and that is using **Calculus**.