Support Vector Machine Python Implementation of Financial Time Series.

Abhay Dodiya

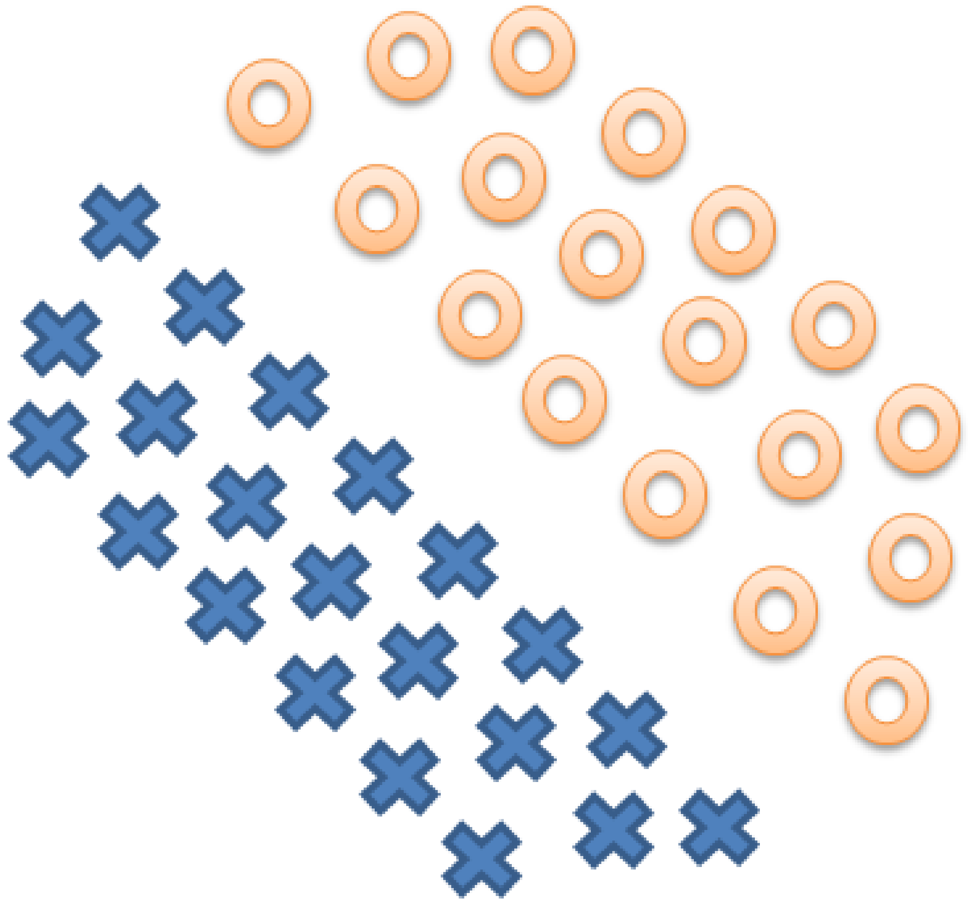
2022 Apr 10

Table of Content:

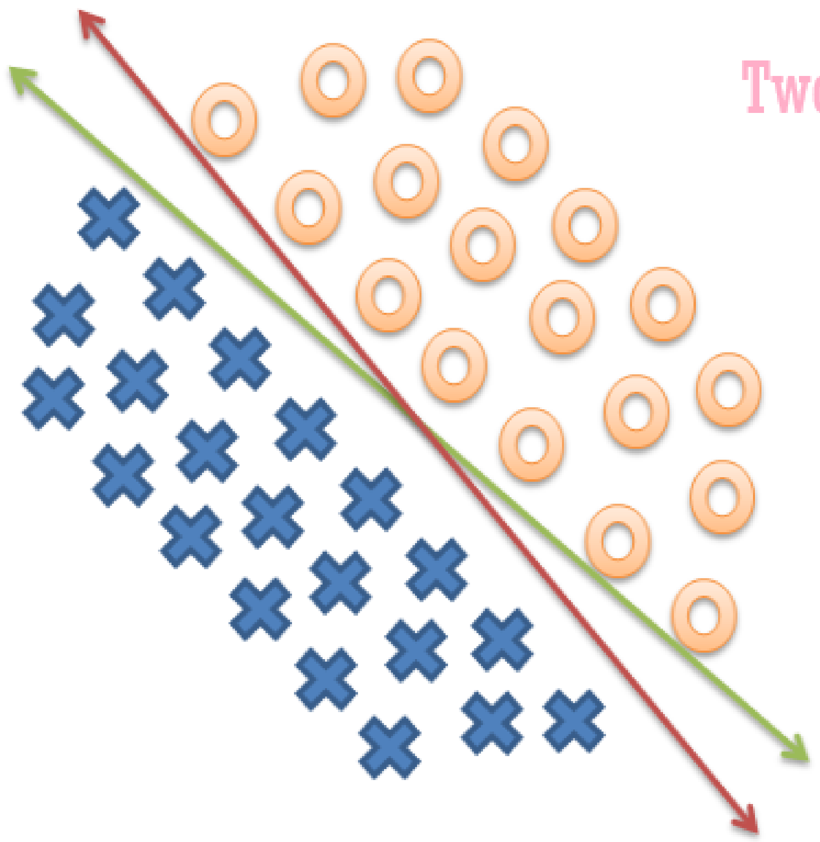
Introduction to SVM

INTRO TO SVM

- SVM is a supervised learning model.
- It means you need a dataset that has been labeled.
- Let's say you have daily returns of financial time series, we can label them either positive return as 1 or negative return as 0.
- It's also known as the classification technique, where it classifies input based on a trained model.



Scatter Plot



Two Seperating Lines to identify best fit.

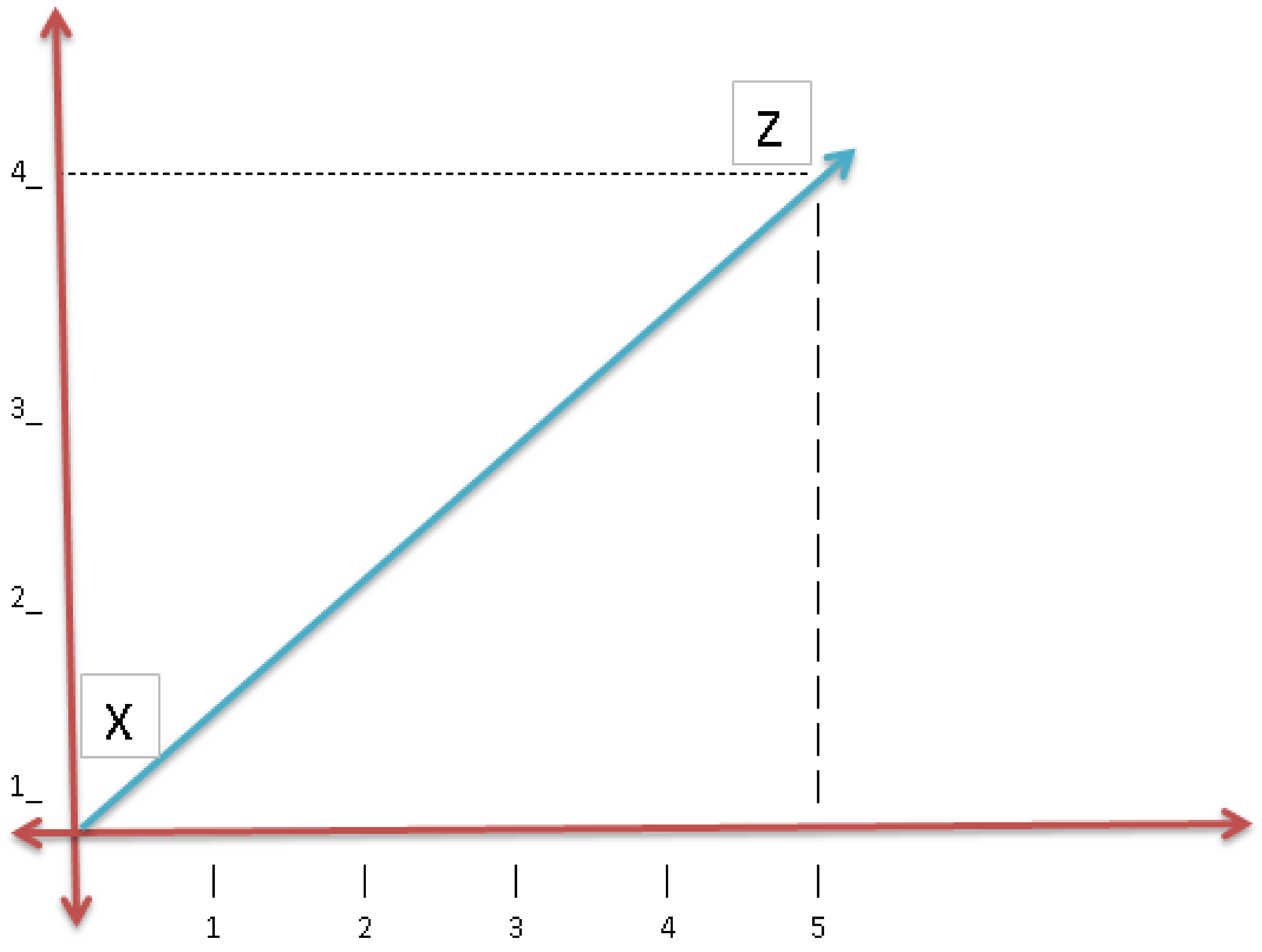
Maths Prerequisite

- In order to Understand the intuition behind SVM, we need to understand two important topics.
  - Dot Products.
  - Lagranges Multiple

- The entire idea begins with a simple understanding of Vector first in order to proceed with Dot Product.

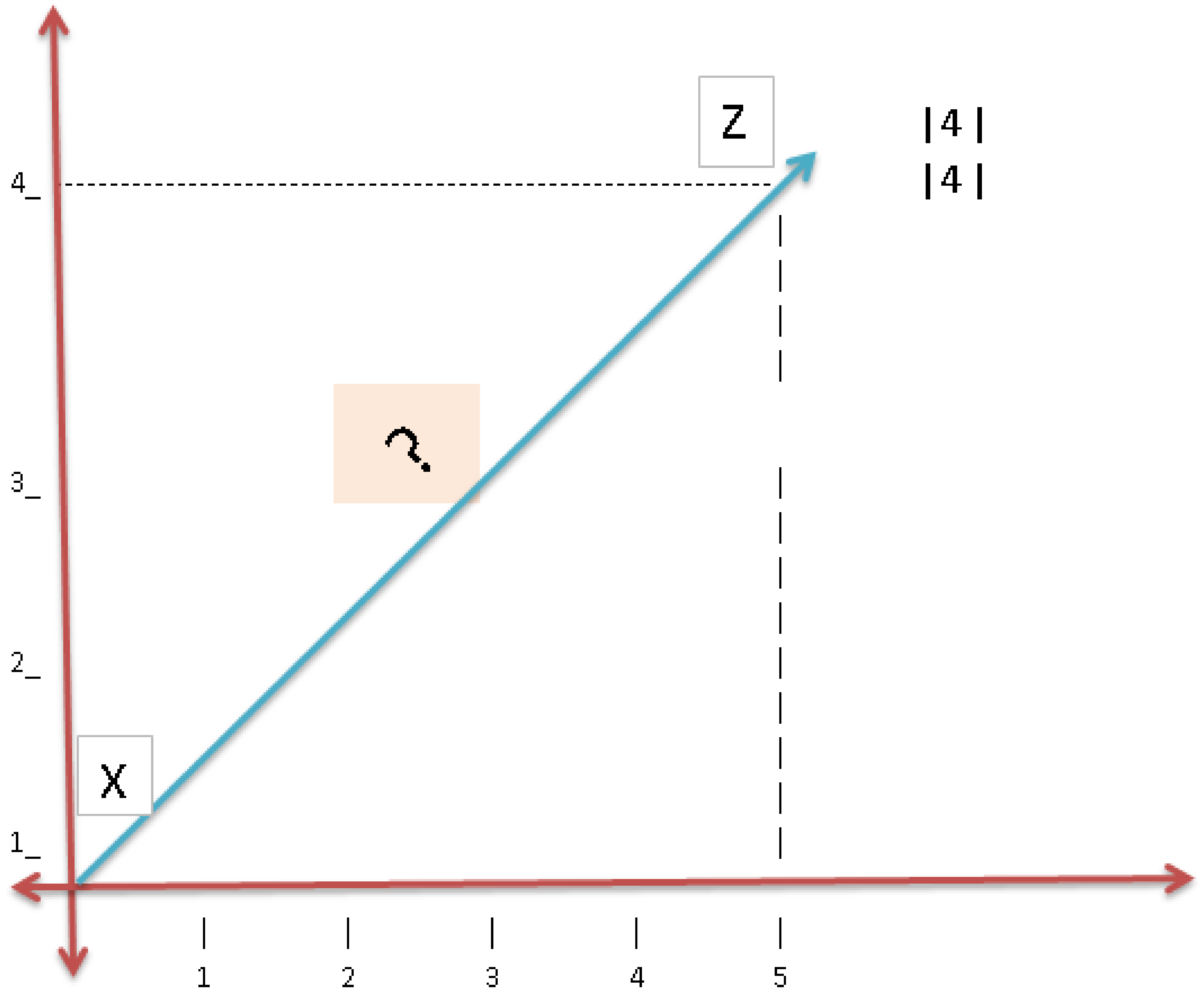
- Vector is nothing but reaching from X point to Z Point in the same direction.

- If you walk the same number of steps in a different direction, then you will not reach Z point but will reach somewhere else.


Z
X
4_
3_
2_
1_
1
2
3
4
5

- In the previous slide, we saw Vector traveling from "X" to "Z".
- In this case distance from "X" to "Z" would be Magnitude.
- Direction would be co-ordinate of the x-axis and y-axis for this magnitude.

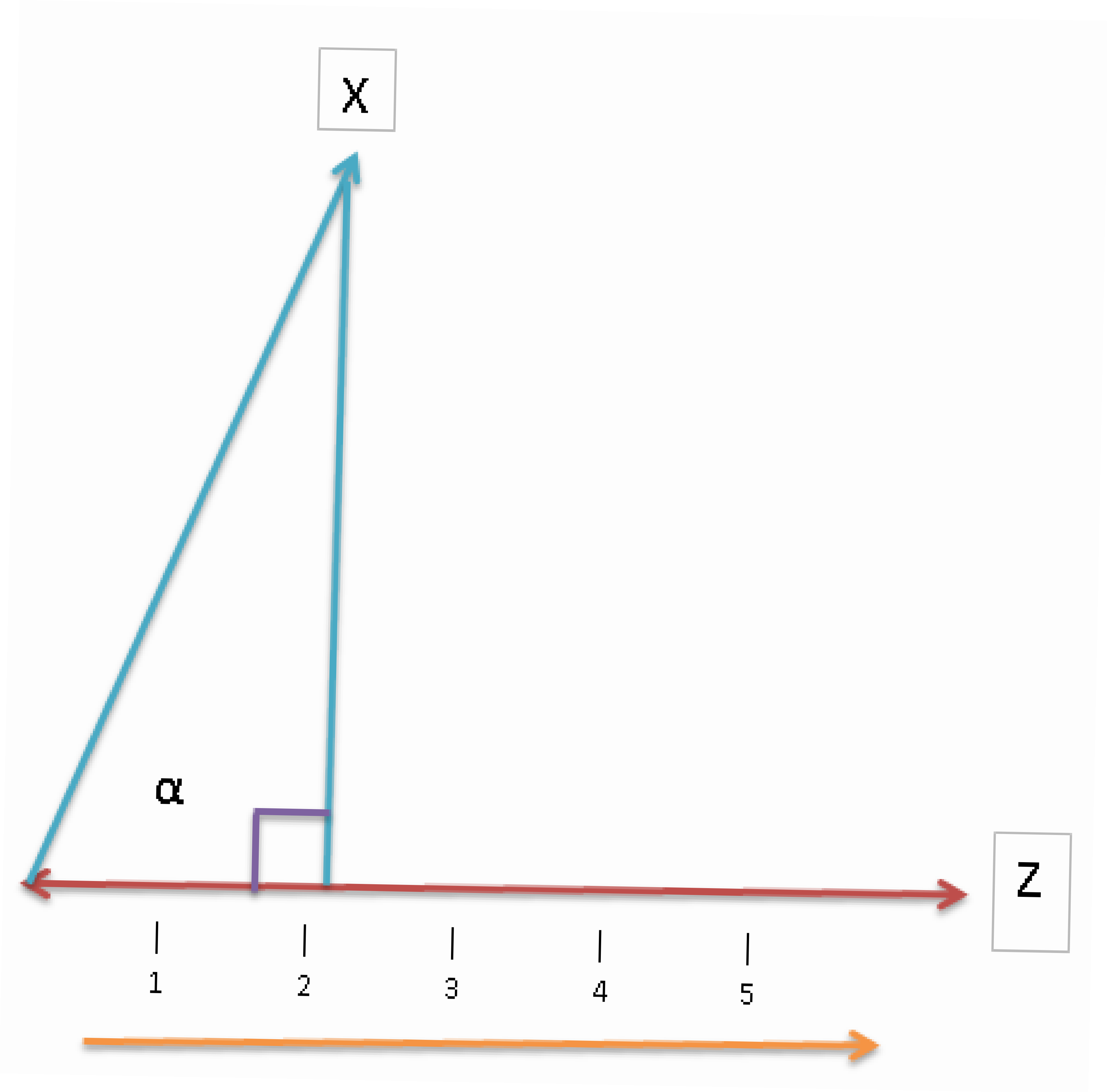
- If we know the direction of the vector, how do we find out the magnitude of that vector?
- In this case, we can revisit Pythagoras' theorem to find out the magnitude.
- Pythagoras theorem: $a^2 + b^2 = c^2$



Z
|4 |
|4 |
4_
3_
2_
1_
?
X
1
2
3
4
5

- Here comes the idea of the dot product, imagine if we have two such vectors pointing in the same direction.

- How much does one vector overlap another vector?


X
α
Z
1
2
3
4
5

- As per the previous slide, there are two vectors X and Z.

- The direction of the vector ‘X’ can be reached by going in the direction of the vector ‘Z’ up to a certain point and subsequently moving in the perpendicular direction.

- This is what the dot product of a vector is all about. It helps us to obtain an accurate measurement of how two vectors are moving in the same direction by telling us how much one vector moves in the direction of the other vector before branching out in a perpendicular direction towards that vector.

- So far we have seen examples of linear separability.
- However, there could be a scenario where we need to work on data that are not linearly separable.
- In such a case, we need to find an optimal solution with constrain in place. This can be solved using Lagrange Multipliers.

- Finding minimum or maximum solutions in the multivariate equation requires a Lagrange multiplier.

- Let's look at an example where we need to minimize cost and get maximum output.

- In a company there are two methods to produce products, we should know the best combination of each method at minimum cost.

- Total Production = 300

- The Constrain would be that we have only two plants to produce this quantity, so the equation would be $G = X + Y = 300$.

- The cost of plant X and Y is as follow X^2 and $2Y^2$. $C = X^2 + 2Y^2$.

- Let's take the partial derivative for the equation with respect to X and Y.

- G(x) = 1
- G(y) = 1
- C(x) = 2x
- C(y) = 4y
- The Lambda here would be a multiplier to solve equation x = λ/2 and y = λ/4.

$$C_x = \lambda g_x$$

$$C_y = \lambda g_y$$

Lagrange

- Solving the first equation of X + Y = 300 with Lambda values of $\lambda/2 + \lambda/4 = 300$.

- We get $\lambda = 400$.

- $x = \lambda/2 = 400/2 = 200$

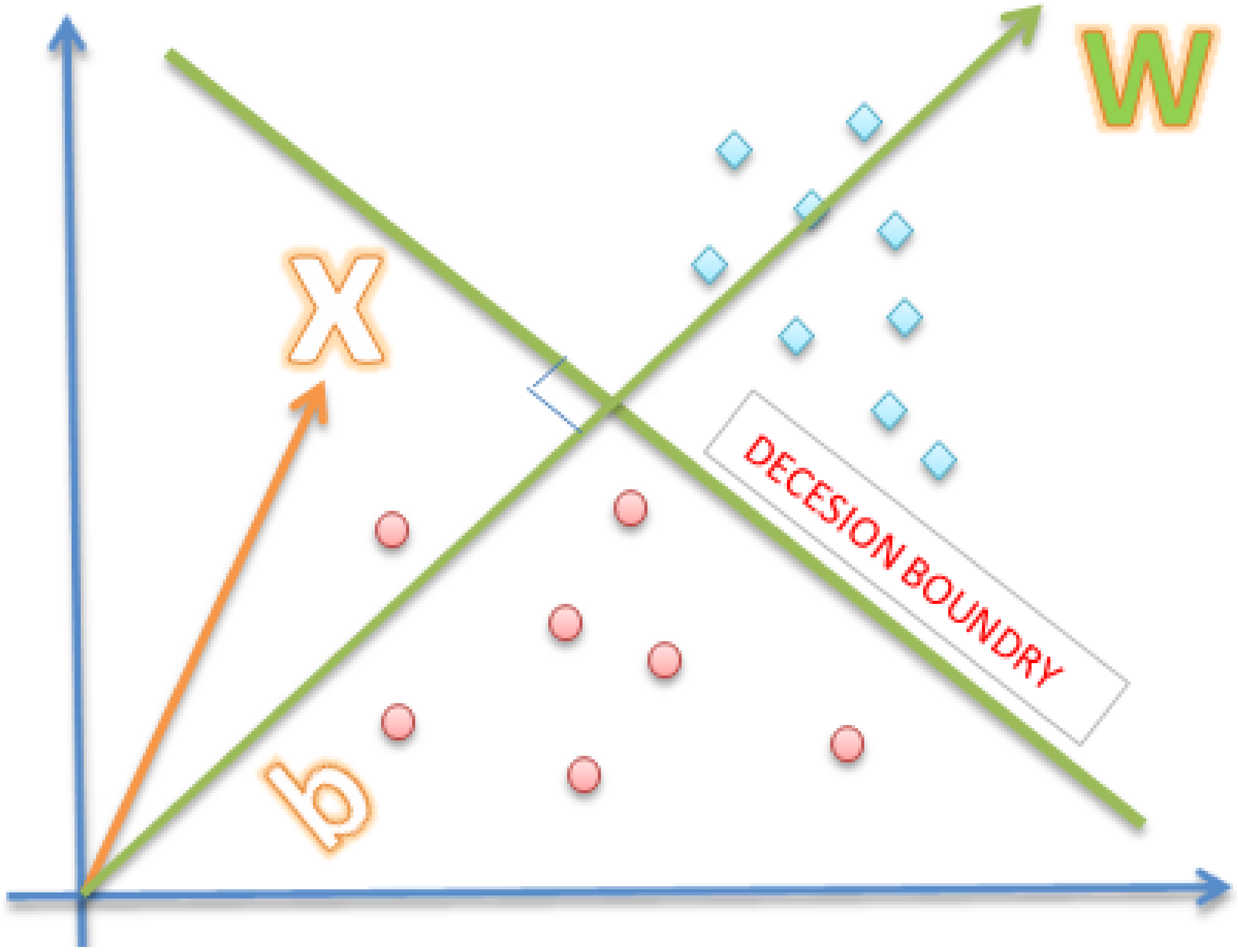
- $y = \lambda/4 = 400/4 = 100$

- To minimize the objective function, we must produce 200 quantities from X plant and 100 quantities from plant Y.

Concept of SVM with Maths.

- As Discussed earlier, SVM is a classification problem, where a dot is placed on a plane and there is a boundary, we need to identify if that dot is above or below the decision boundary.

- In this case, "X" is a new a dot placed on the plane and there is already one decision boundary, we need to use Dot Product to identify if a dot is placed above / below or on a boundary.



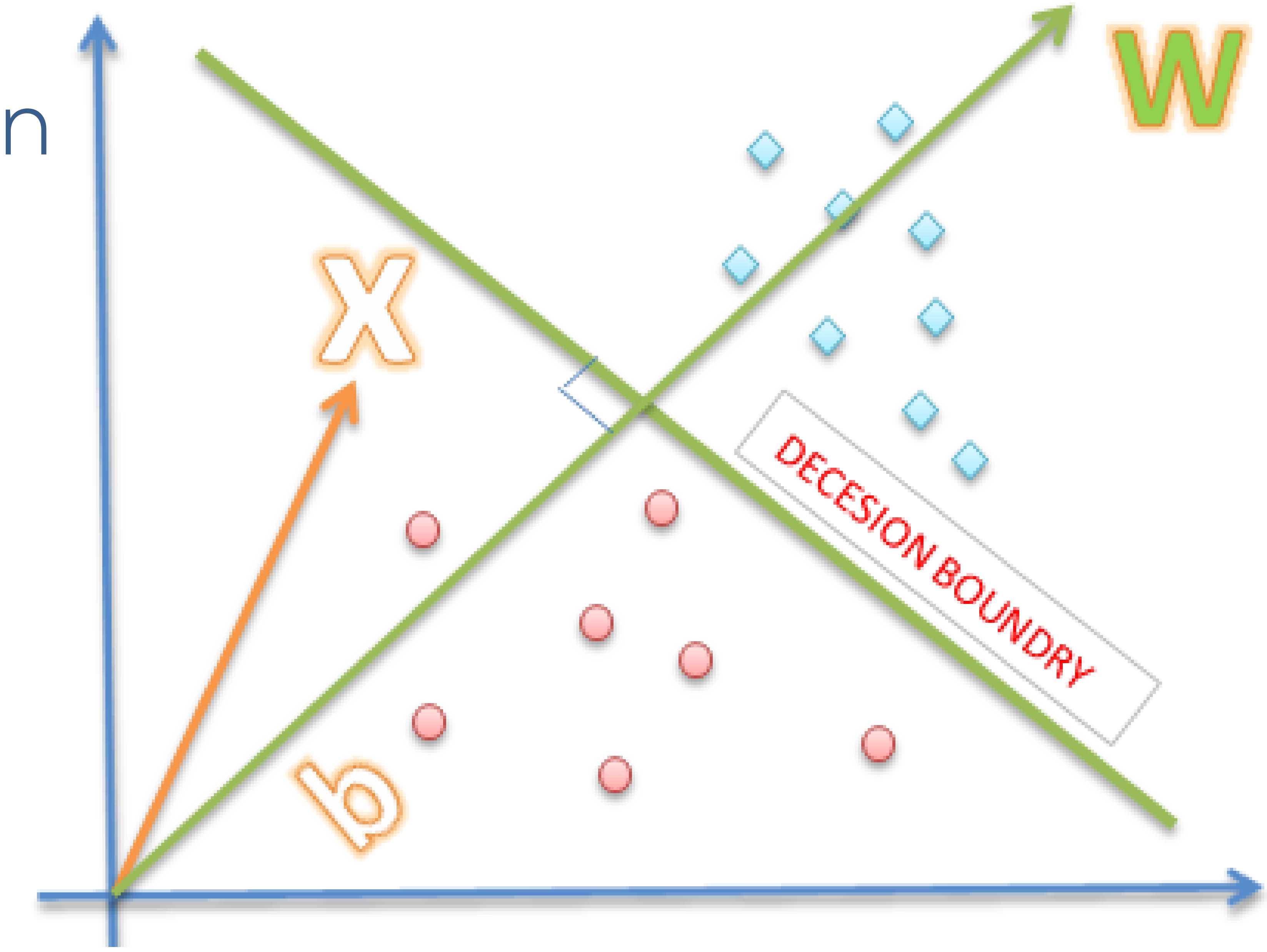
- "W" here is a perpendicular to the boundary because X can lie anywhere on the decision boundary, but when we do dot product X and B will be on the same line and it's perpendicular and based on the magnitude of X it'll be

- On decision boundary when

  $X * W = b$
- or Positive Sample when :

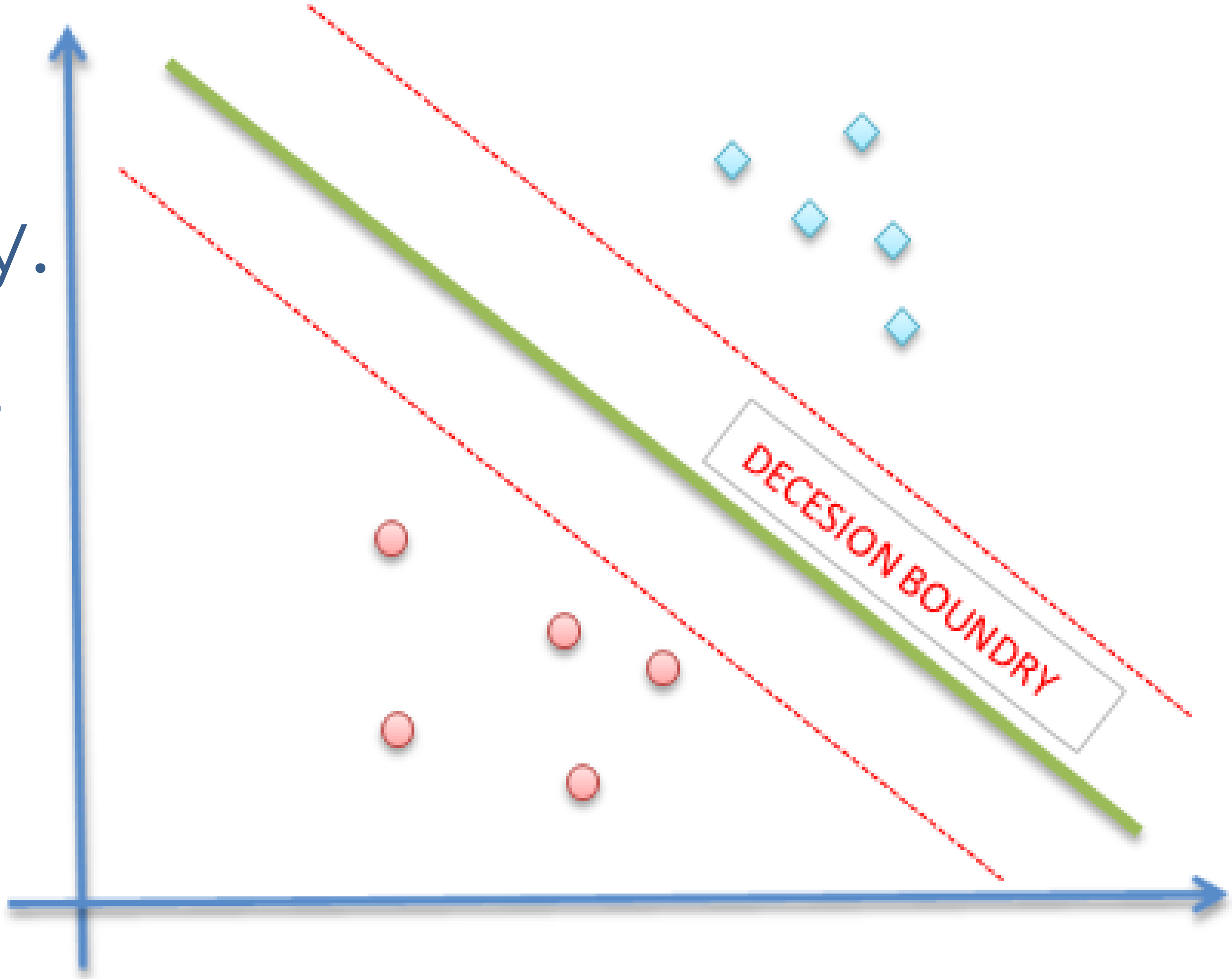
  $X * W > b$
- or Negative Sample when :

  $X * W < b$



- So far we have discussed a single line as a decision boundary. however, we need to keep some margin on both sides of the decision boundary.

- The red lines on both sides are margin to the decision boundary. Assume margin width of at least 1. In that case, the equations are



- Positive Samples :

  $X * W - b > 1$

- Negative Samples :

  $X * W - b < -1$

- Let's create a variable Ya = 1 for positive samples and Ya = -1 for negative samples, in order to remove inequality and have only one equation.

- If we perform the following operations we'll get one equation:

  +1 * (X * W - b) > 1 * +1 for Positive Samples.
  -1 * (X * W - b) < -1 * -1 for Negative Samples.

- Now, both of the equation leads to only one equation.
- Ya * (X * W -b) > 1.

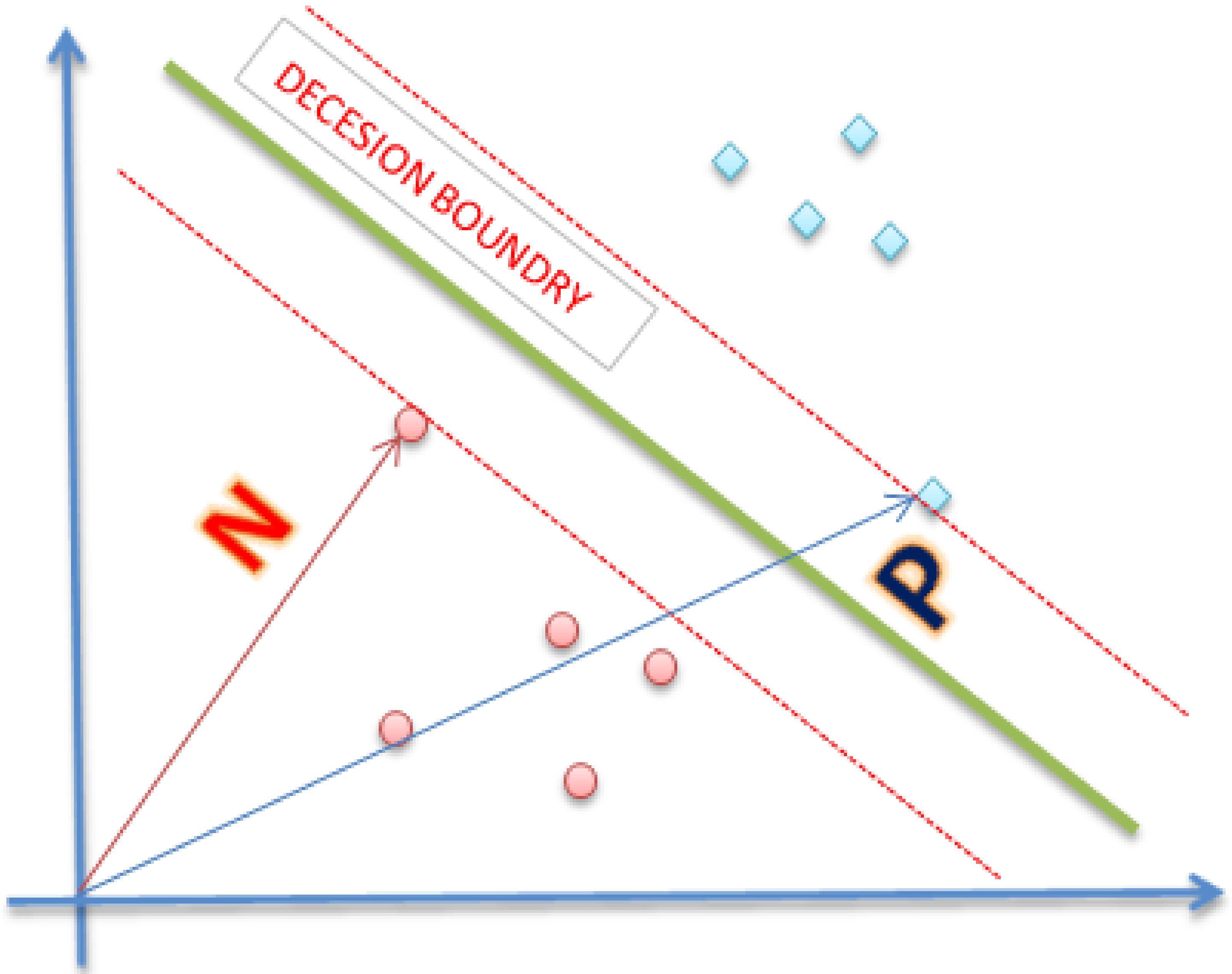
- Finally, the Output equation would be as follows.

  Ya * (X * W -b) -1 > 0 for all samples and,
  Ya * (X * W -b) -1 = 0 when point is on Decision Boundry.

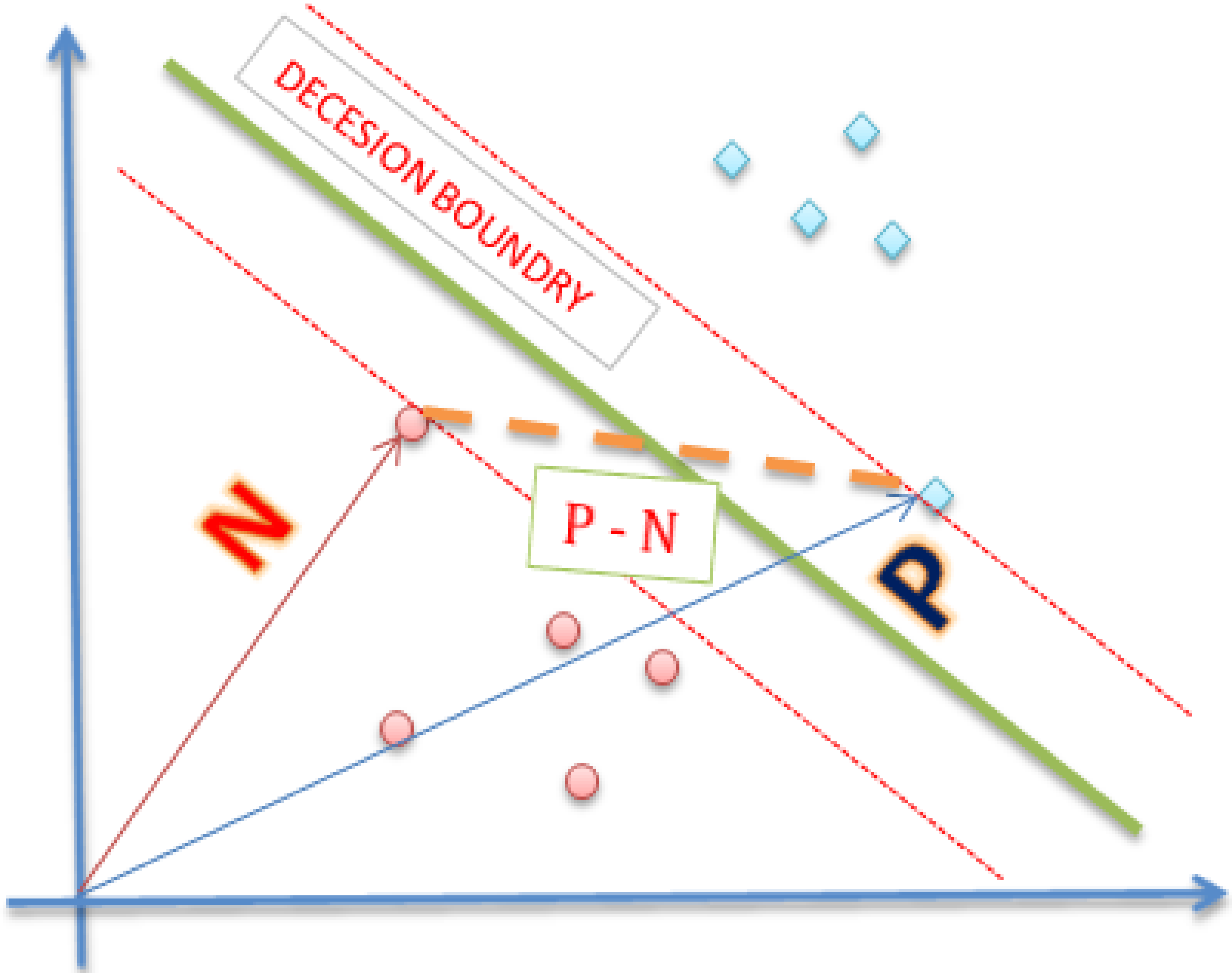
- The next step is the Optimization problem, Since this problem of maximum margin.

- We are going to deal with Maximum Width along the decision boundary, better the margin, better the projection.

- In the below figure, where we take two points of each sample which lies on decision boundary.



- Now in order to identify perfect width, let's join these two vectors, and vector arithmetic it's simply (P-N).



- In the previous slide, since it's diagonal distance between "P" and "N" it can lie anywhere on "N".
- So, we need a line that is perpendicular to the decision boundary.
- Taking projection of the vector along the perpendicular ("W" is a unit vector).
- Width = (P-N) * W / Magnitude(W).

- So, If $Y=1$ and $X=P$ for $Ya\ (X * W - b) > 1$ for all positive samples, we get $P = (1+b)/W$.

- For all negative samples, we get $N = (-1+b)/W$.

- $P-N = (1+b)/W - (-1+b)/W$
    - Maximize $= 2/W$.

- In order to Minimize the above equation, we can simply inverse the equation, the new equation would be $W/2$.

- We know "W" is perpendicular to the decision boundary but we don't know "W".

- We can maximize "W" to infinity but we can't do that as we need to satisfy certain conditions which we have defined previously and that will act as a constraint.
  - Ya (X * W -b) - 1 > 0 all points. &
  - Ya (X * W -1) = 0 for all points on the margin.

- Now, we have to do two things simultaneously i.e. maximize margin and also keep constraints in place.

- In order to do this, we will use Lagrange Multiplier which is one of the most used theorems in Calculus, which we discussed earlier.

- Instead of W/2, we will use W^2 / 2 the reason being in the convex function, we have only one minima. however, in the non-convex function, we have many local minima.

- Our Lagrangian equation would be as follow.

$$W^2 / 2 - \Sigma\, C(i)\, [Ya * (W * X(i) - b) - 1]$$

- If we differentiate L w.r.t W we derive the following equation.

$$\frac{W^2}{2} - \sum C^i * [\, Y^a * (\, X^i * W - b\,) - 1 \,]$$

$$\frac{W^2}{2} - \sum C^i * Y^a * X^i * W - C^i * Y^a * b - C^i * 1$$

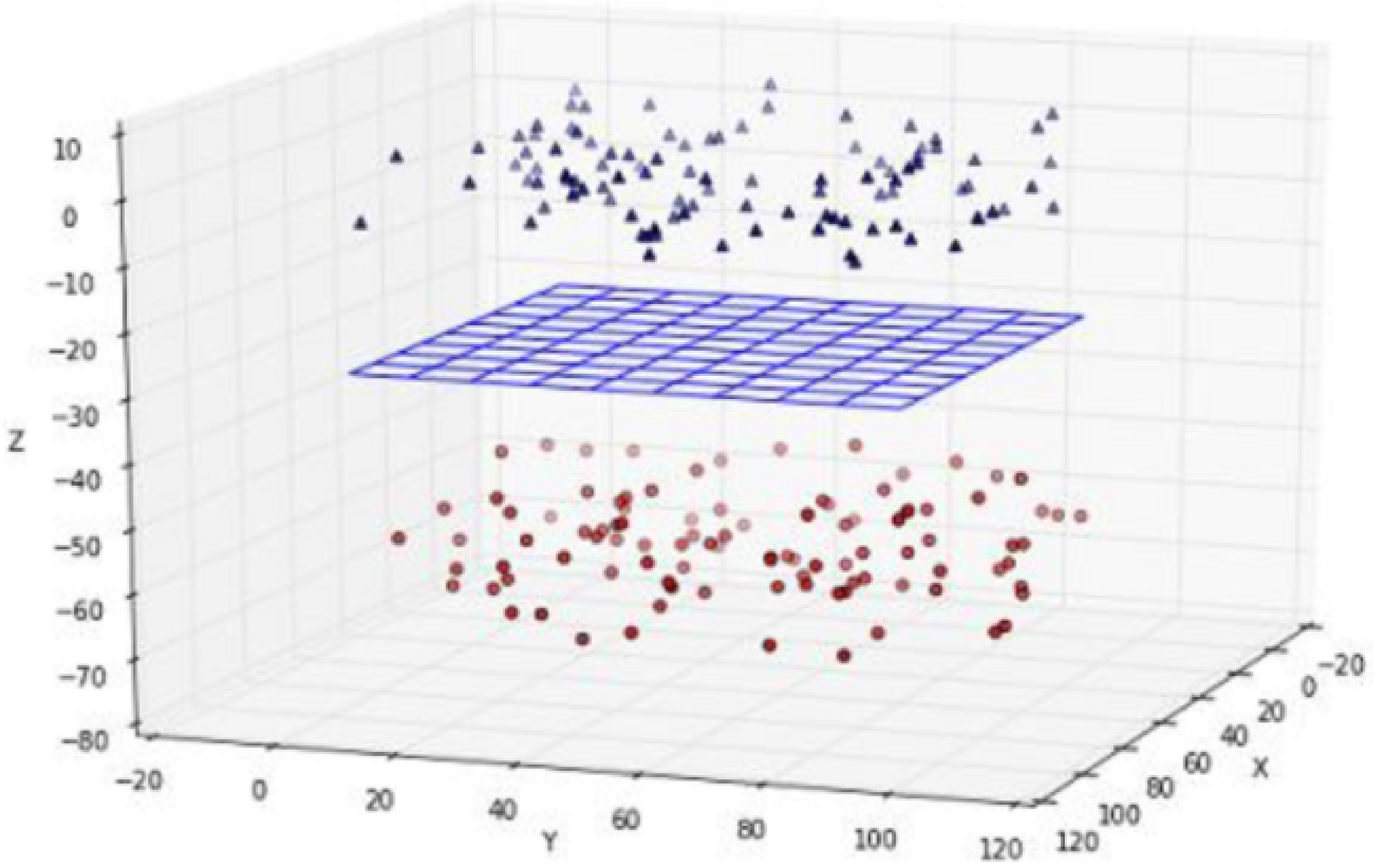
$$\frac{1}{2} * \frac{d}{dW} W^2 - \frac{d}{dW} C^i * Y^a * W * X^i - \text{Constants}$$

$$W = \sum_{i=0}^{i} C * Y^a * X^i$$

- So, the Final equation would be a linear combination of data.

- What if our data is non-linear, as shown below example. In such a scenario we use something called a "kernal-trick".

- Kernal Trick transforms the data into a new dimension, where you can separate them.

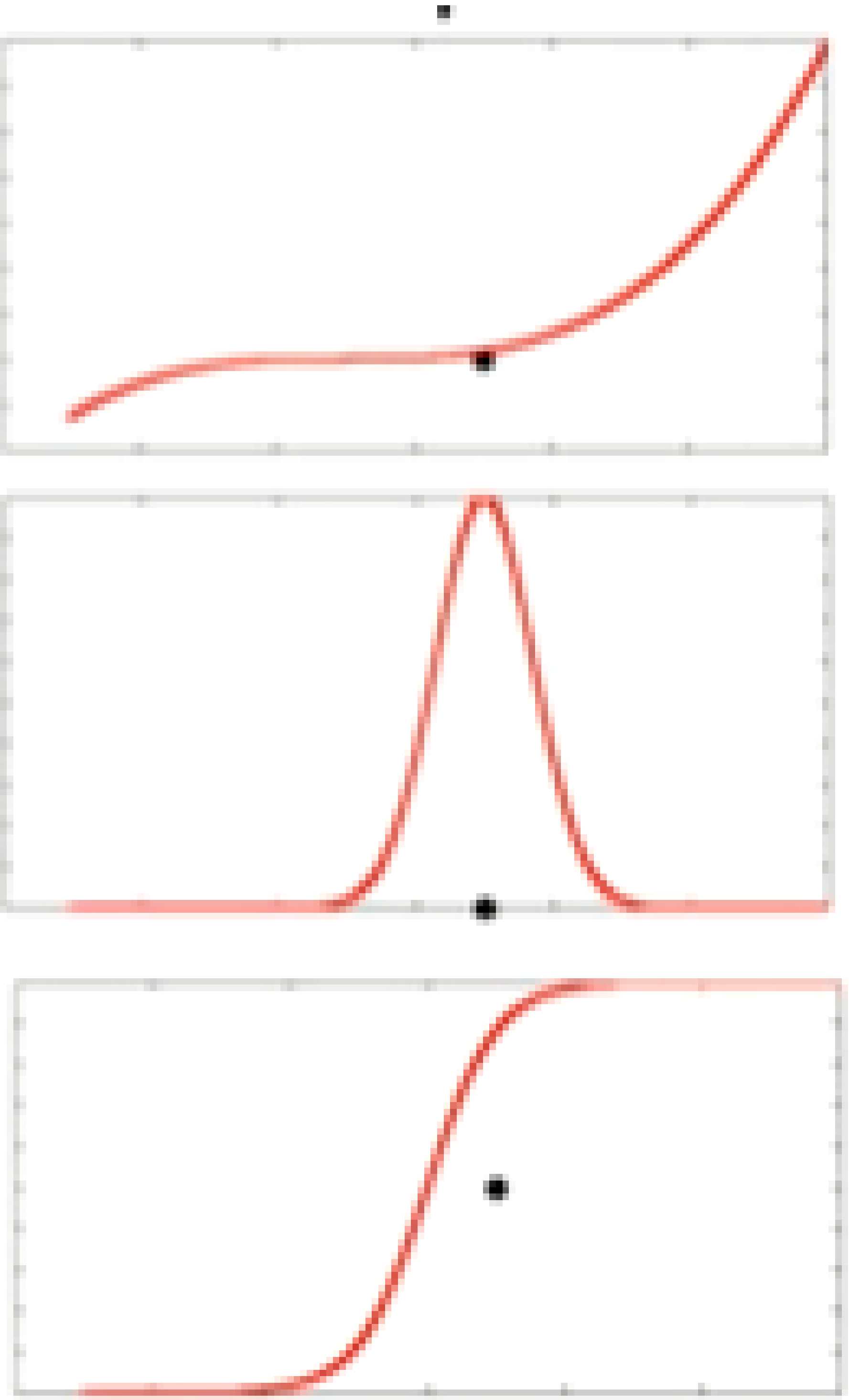


- There are different kernels in SVM, which basically transfer into a higher dimension, for them to be linearly separable.

  - Polynomial Kernel.
  - Sigmoid Kernel.
  - Radial Basis Function (RBF) Kernel.

Concept of SVM

- Polynomial $K(a,b) = (1 + \sum_j a_j b_j)^d$
- Radial Basis Functions
  $$K(a,b) = \exp(-(a-b)^2/2\sigma^2)$$
- Saturating, sigmoid-like:
  $$K(a,b) = \tanh(ca^T b + h)$$
- Many for special data types:
  - String similarity for text, genetics



Python Implementation – Simple Example.

- Financial time series data for "NIFTY" for the last five years was downloaded from yahoo finance.
- Added, two technical indicators. namely "RSI" and "ADX" for trend and momentum. They are used as X Variable to help us predict the next move.
- If the daily return is positive then assign 1 and if it's negative then assign 0. This will be our "y" variable.

- The first step is to split the dataset between train and test, we are using 80% data as training data.

- We'll sklearn library SVC (Support Vector Classification) to help us perform support vector machine.

- Finally, We can see using accuracy metrics, the outcome is 54%, i.e. it has predicted data correctly 54% of the time.

- This was just an example from an illustration perspective.

- We can further fine-tune our X variables and also performed further optimization, in order to get better accuracy.

Thank You