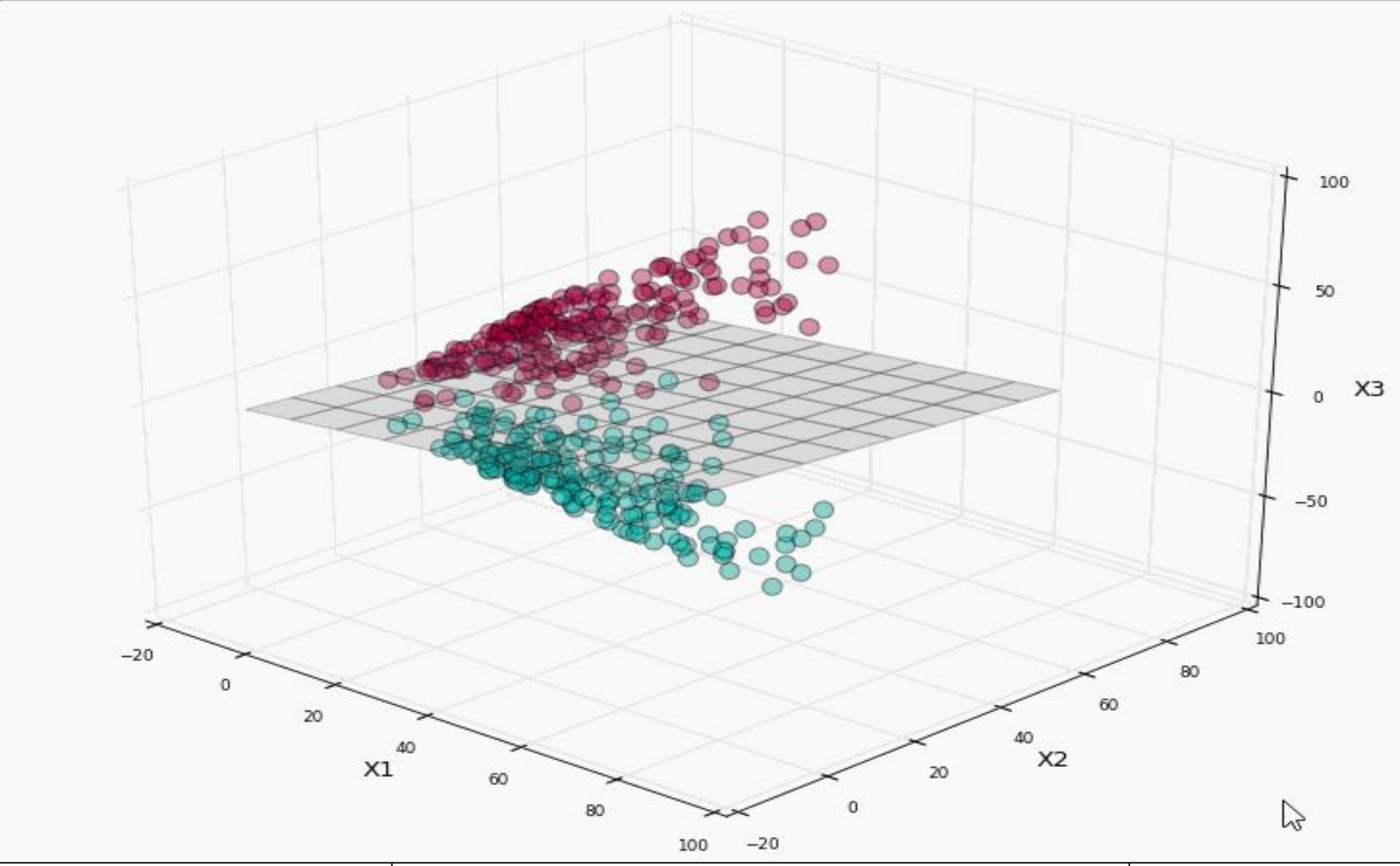
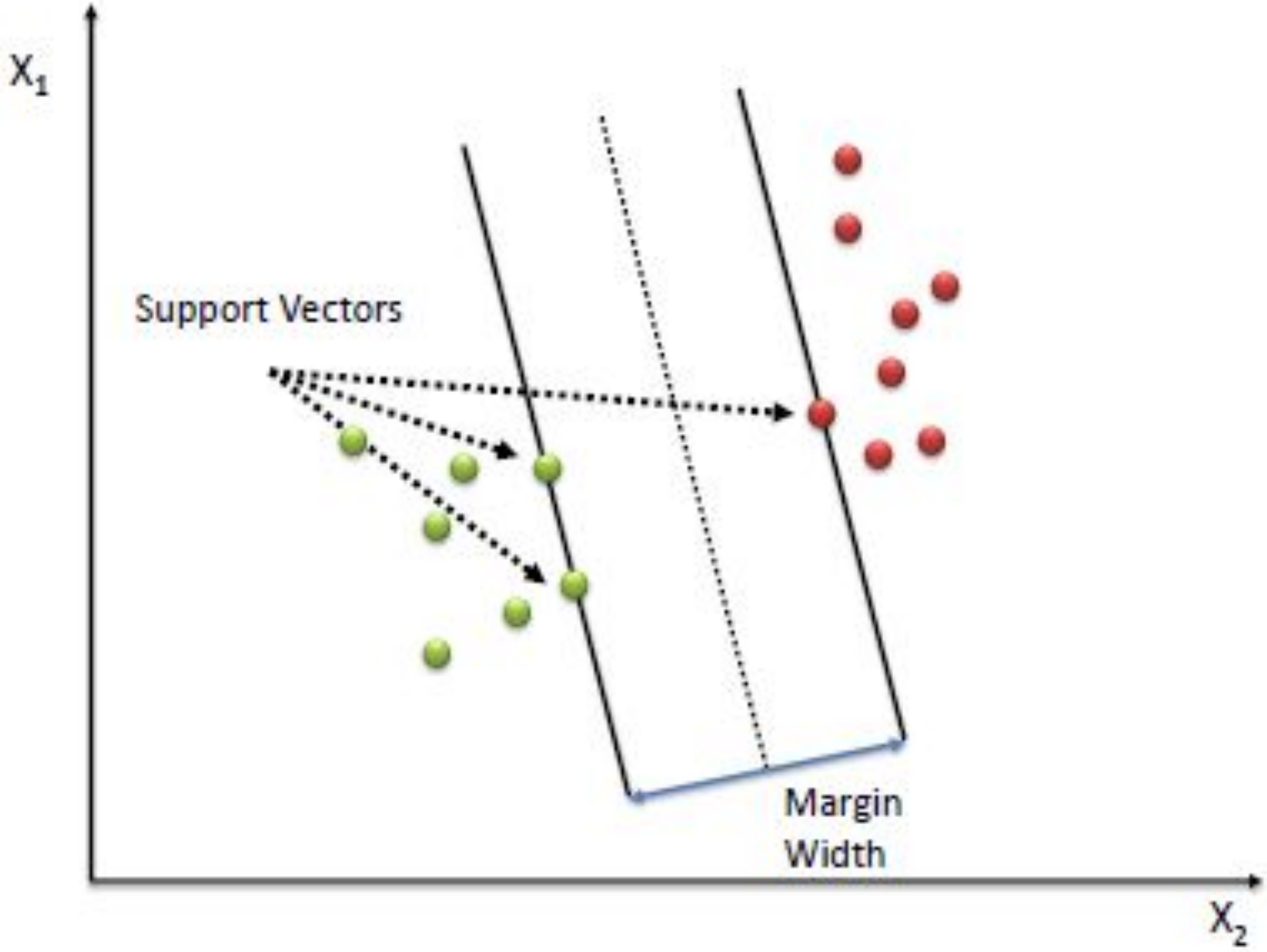


Dr. Nilina Bera | CSE(DS) | HITK

Support Vector Machines (SVM): Maximum margin hyperplanes, Linear SVM: and kernels. separable case, non-separable case, Non-linear SVM

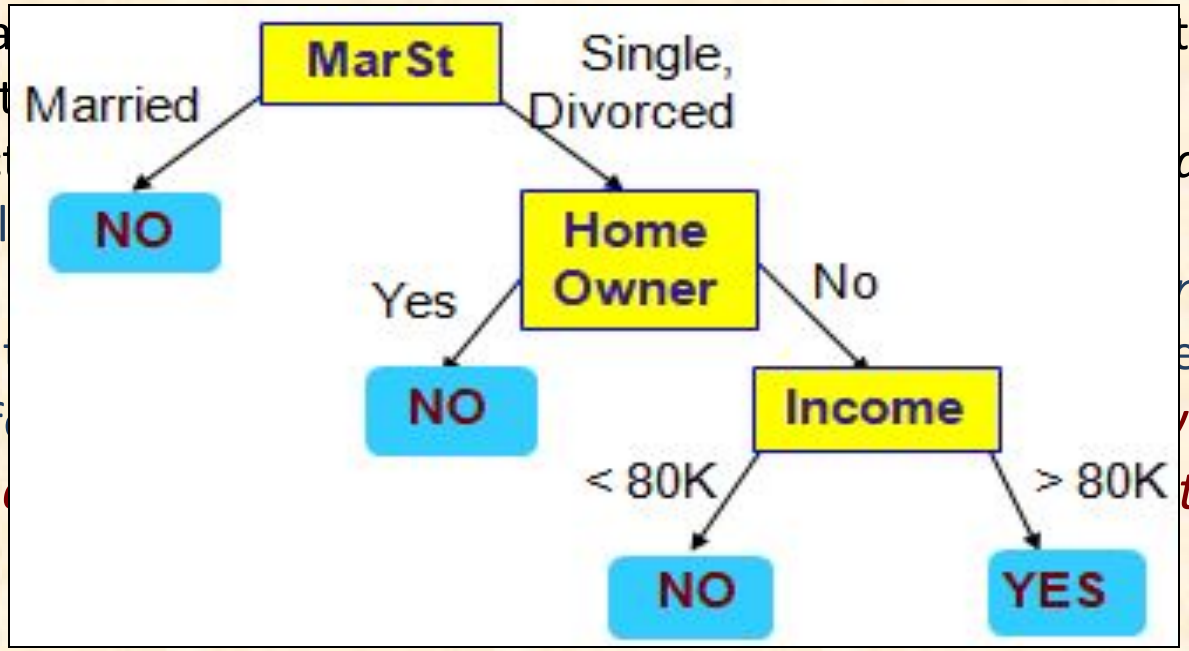

X_1
Support Vectors
Margin
Width
X_2

- Support vector machines (SVMs) are a set of related *supervised learning* methods that analyze data and recognize patterns, used for classification and regression analysis.
- The original SVM algorithm was invented by Vladimir Vapnik and the current standard incarnation (soft margin) was proposed by Corinna Cortes and Vladimir Vapnik.
- The standard SVM is a non-probabilistic *binary linear classifier*, i.e. it predicts, for each given input, which of two possible classes the input is a member of.
- Since an SVM is a classifier, then given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other.

Two keywords: Supervised Learning methods, Binary Linear Classifier

Supervised Learning

Supervised lea…tion that maps an input to an out…
It infers a funct…*aining examples*.
In supervised l… object (typically a vector) and a …*nal*). A supervised learning algori…ed function, which can be used f…*ill allow for the algorithm to …tances*.



Training Set

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
11	No	Married	55K	?
12	Yes	Divorced	80K	?
13	Yes	Single	110K	?
14	No	Single	95K	?
15	No	Married	67K	?

Test Set

Binary Linear Classifier

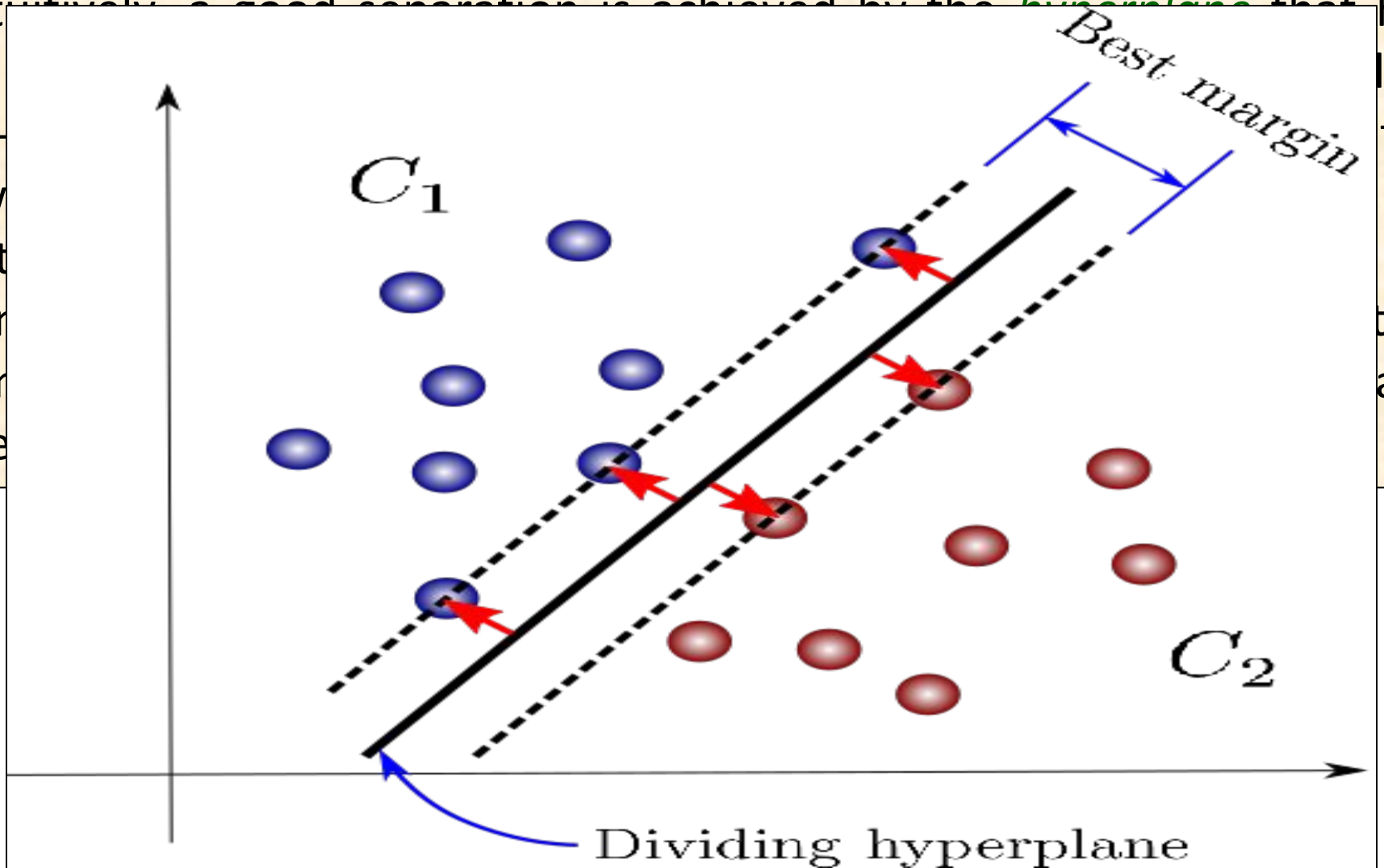
A linear classifier makes a classification decision for a given observation based on the value of a linear combination of the observation's features. In a ``binary'' linear classifier, the observation is classified into one of two possible classes using a linear boundary in the input feature space.

Binary classification, where the goal is to predict a binary-valued target. Here are some examples of binary classification problems:

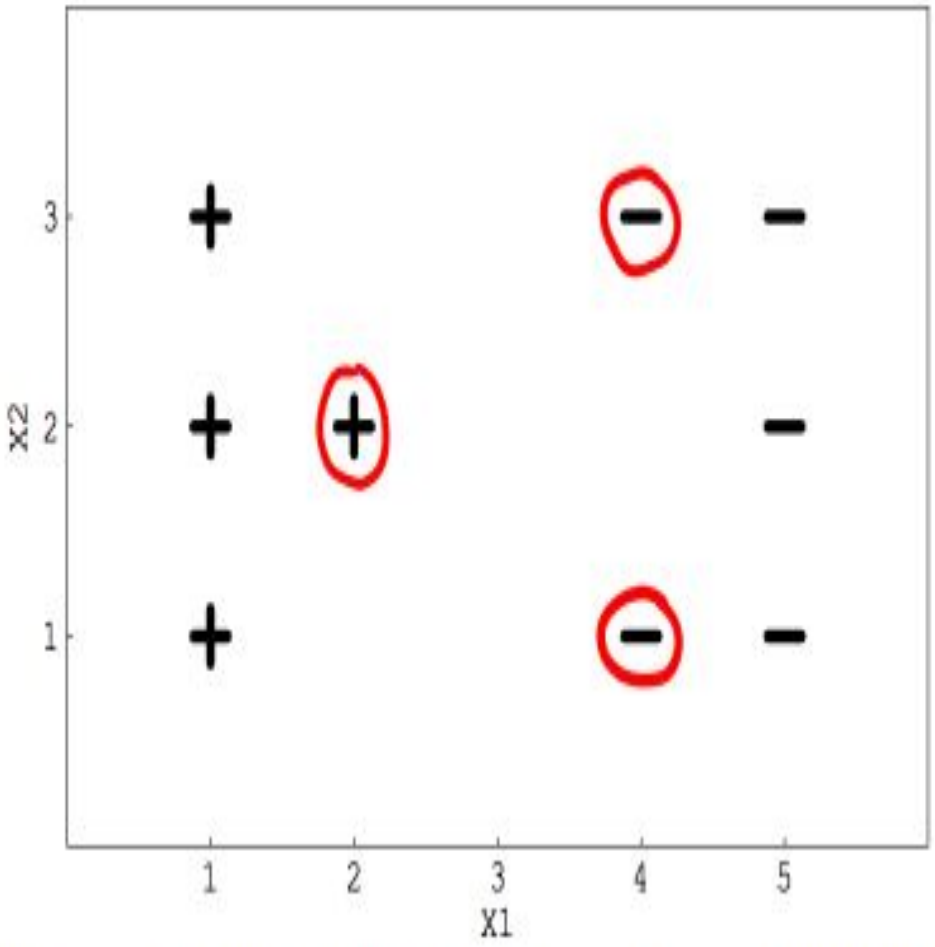
- You want to train a medical diagnosis system to predict whether a patient has a given disease. You have a training set consisting of a set of patients, a set of features for those individuals (e.g. presence or absence of various symptoms), and a label saying whether or not the patient had the disease.
- You are running an e-mail service, and want to determine whether a given e-mail is spam. You have a large collection of e-mails which have been hand-labeled as spam or non-spam.
- You are running an online payment service, and want to determine whether or not a given transaction is fraudulent. You have a labeled training dataset of fraudulent and non-fraudulent transactions; features might include the type of transaction, the amount of money, or the time of day.

Support vector machines (SVMs)

- A support vector machine constructs a *hyperplane* or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks.
- Int… …has
the … …lass
(so- …the
low
- Lot … we
war … …ting
fro… …any
que



Suppose you are using a Linear SVM classifier with 2 class classification problem. Now you have been given the following data in which some points are circled red that are representing support vectors.



1) If you remove the following any one red points from the data. Does the decision boundary will change?

A) Yes

B) No

How Support Vector Machine Works ?

Applications of Support Vector Machine

Face detection

Text and hypertext categorization

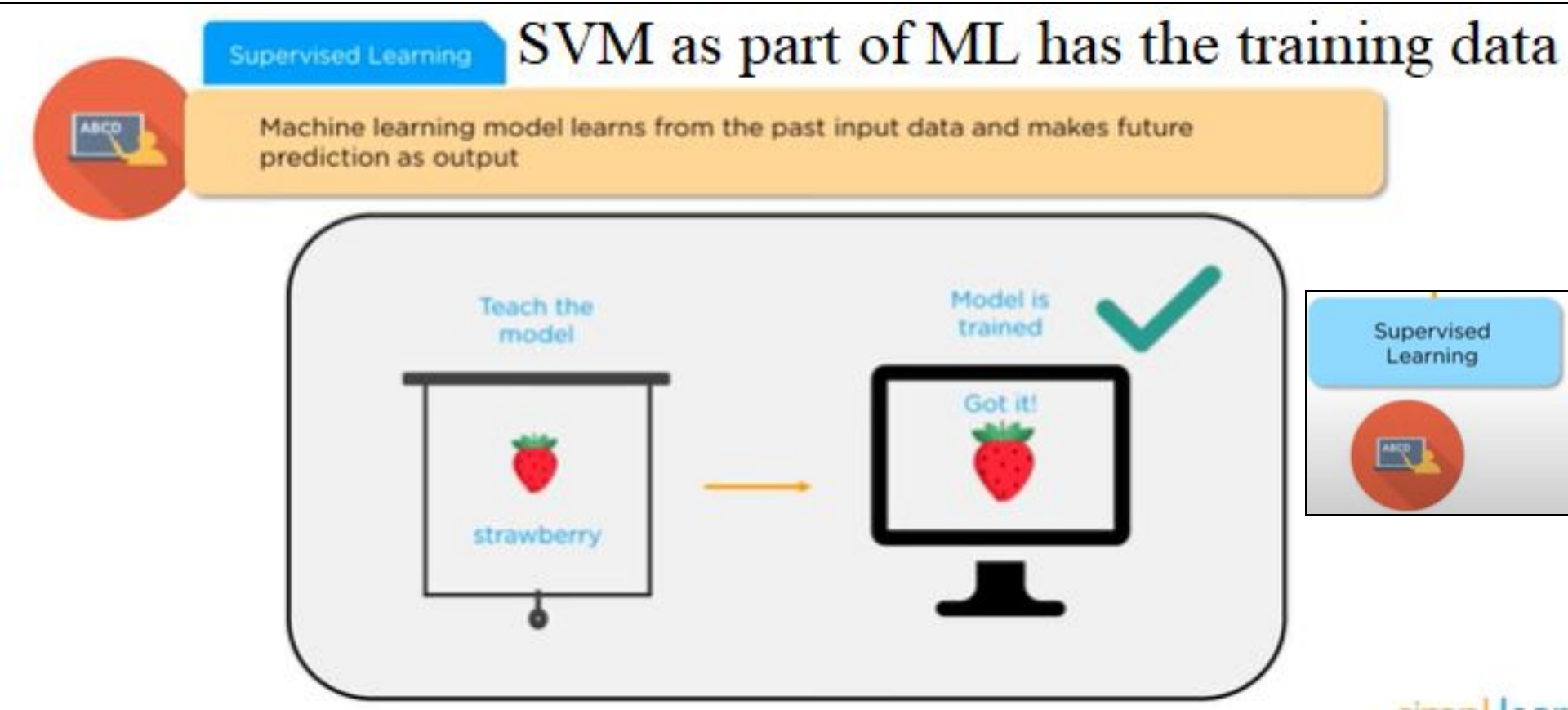
Classification of images

Bioinformatics

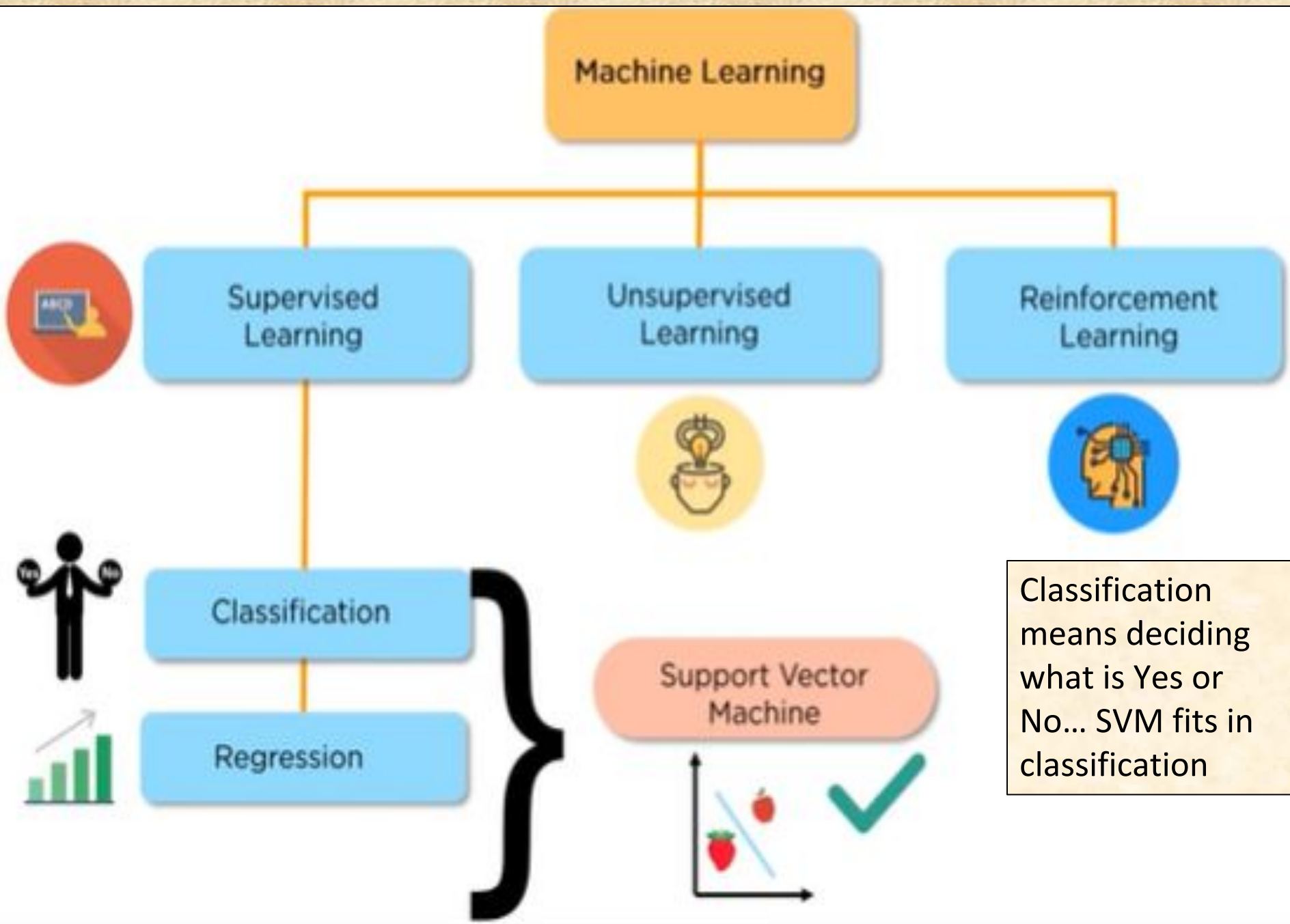
How Support Vector Machine Works ?

A support vector machine transforms training data into a higher dimension, where it finds a hyperplane that separates the data by class using essential training tuples called *support vectors*.

The above is still a high level definition. We are looking for answers to questions like: Why SVM? What is SVM? Understanding SVM... Advantages of SVM...



Where does Support Vector Machine fit in?



Classification means deciding what is Yes or No… SVM fits in classification

Where does Support Vector Machine fit in & Why SVM?

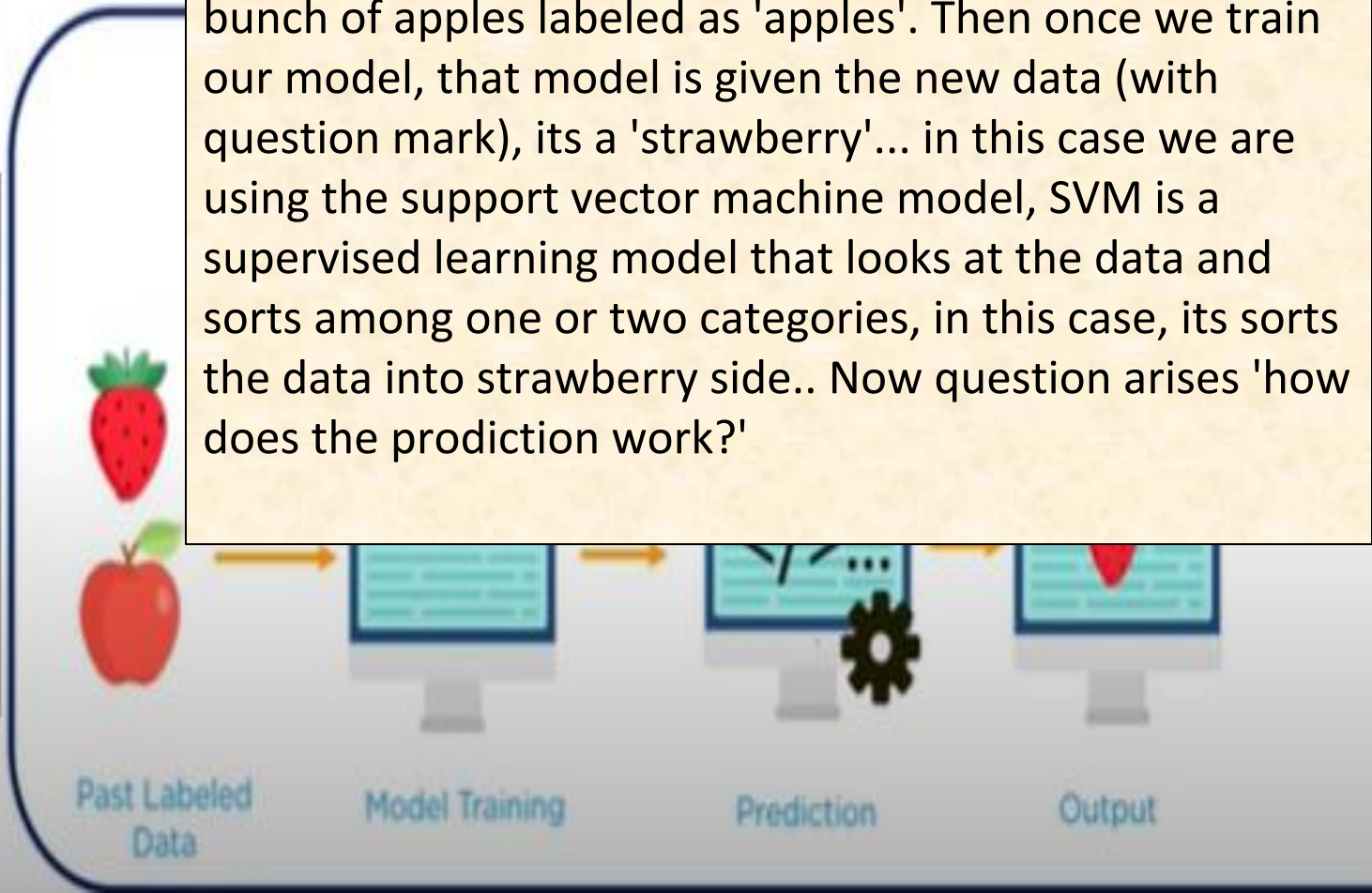
Let’s try to answer this question by real life example: In a fruit shop, both Apples & Strawberry's are racked…

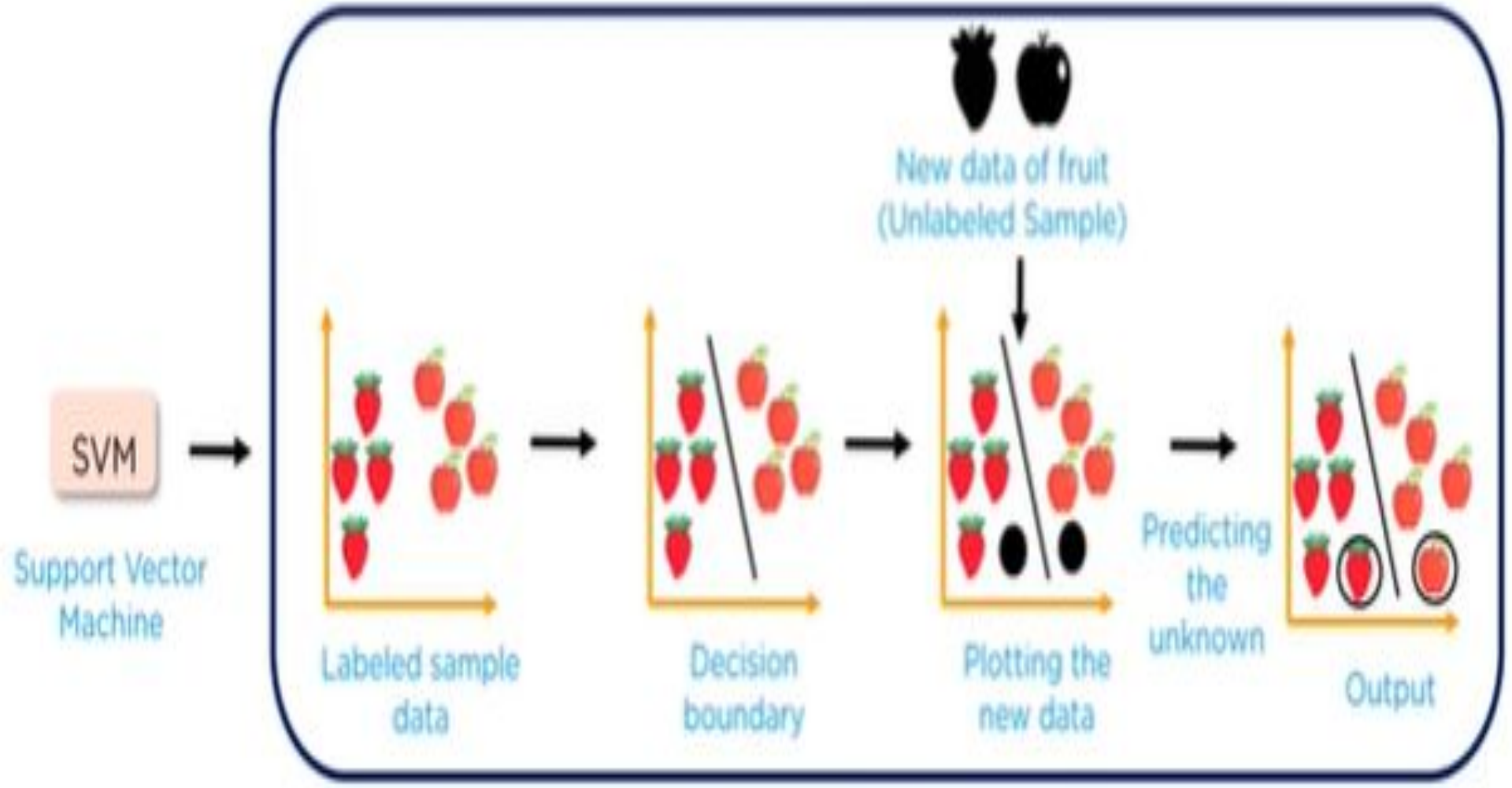
My Son picking up a strawberry and is asking me “Ma, is this an apple? Looks different”… well, his knowledge does not cater to strawberry, it’s a new fruit that’s he has seen for the first time & he wants to classify it! Thus, I taught him to identify the new fruit with name ‘Strawberry’. Question arises: Why don’t we build a model that would predict unknown data and classify correctly ????

Why Support Vector Machine?

Why not build a model which can predict an unknown data??

We are going to look into sweet strawberries or crispy apples and we are going to label 'what exactly this fruit is' -- to do this we already have the data fed in, we have the training instances, i.e., a bunch of strawberries and we have labeled as strawberries and we also have a bunch of apples labeled as 'apples'. Then once we train our model, that model is given the new data (with question mark), its a 'strawberry'... in this case we are using the support vector machine model, SVM is a supervised learning model that looks at the data and sorts among one or two categories, in this case, its sorts the data into strawberry side.. Now question arises 'how does the prodiction work?'

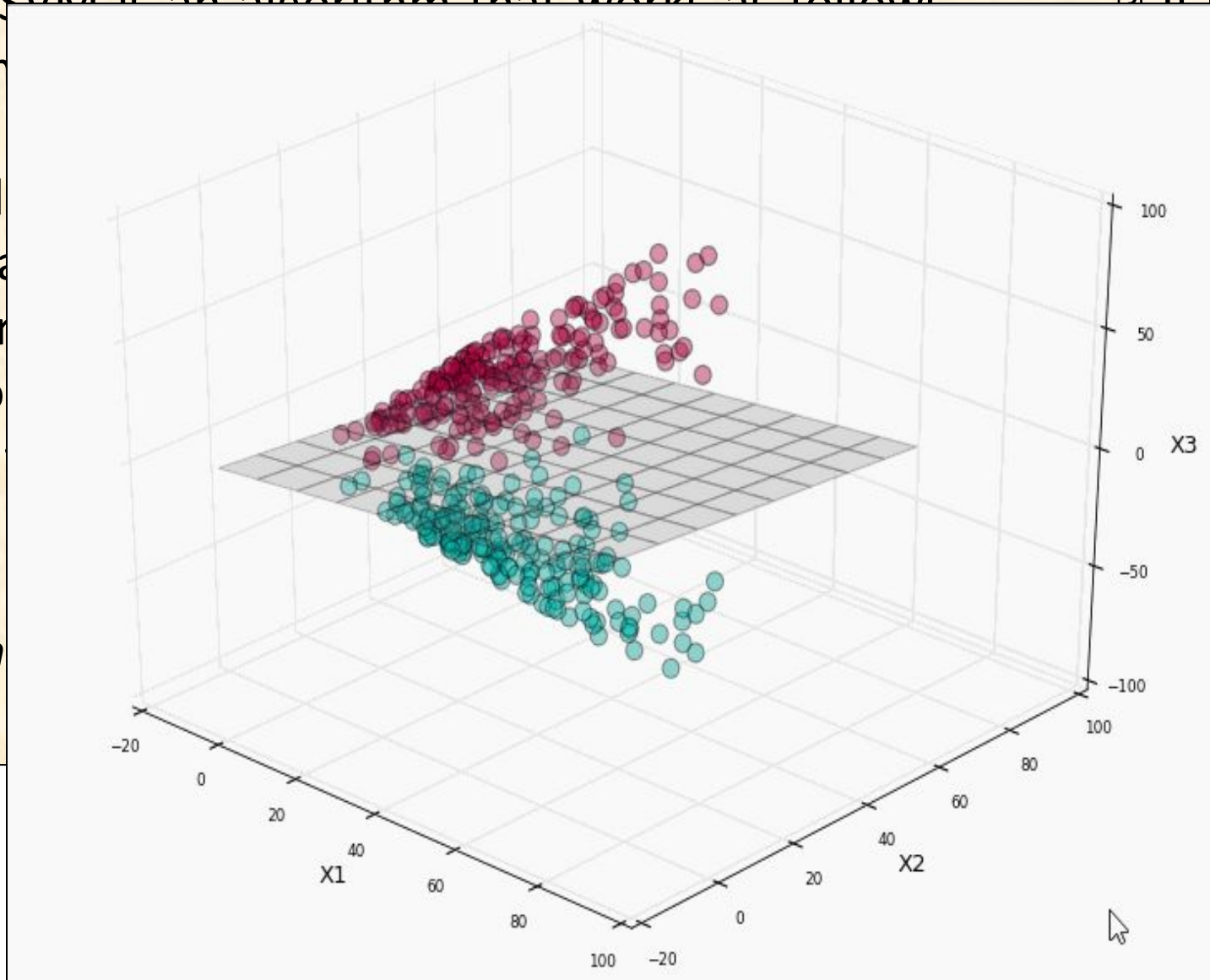




We have taken labelled sample data, i.e., training data, lets draw a line down the middle between the two groups, a decision boundary. This split allows us to take new data and allows us to put new data, in this case, an apple and a strawberry and place them in the appropriate group based on which side of the line they fall in. This way we can predict the unknown.

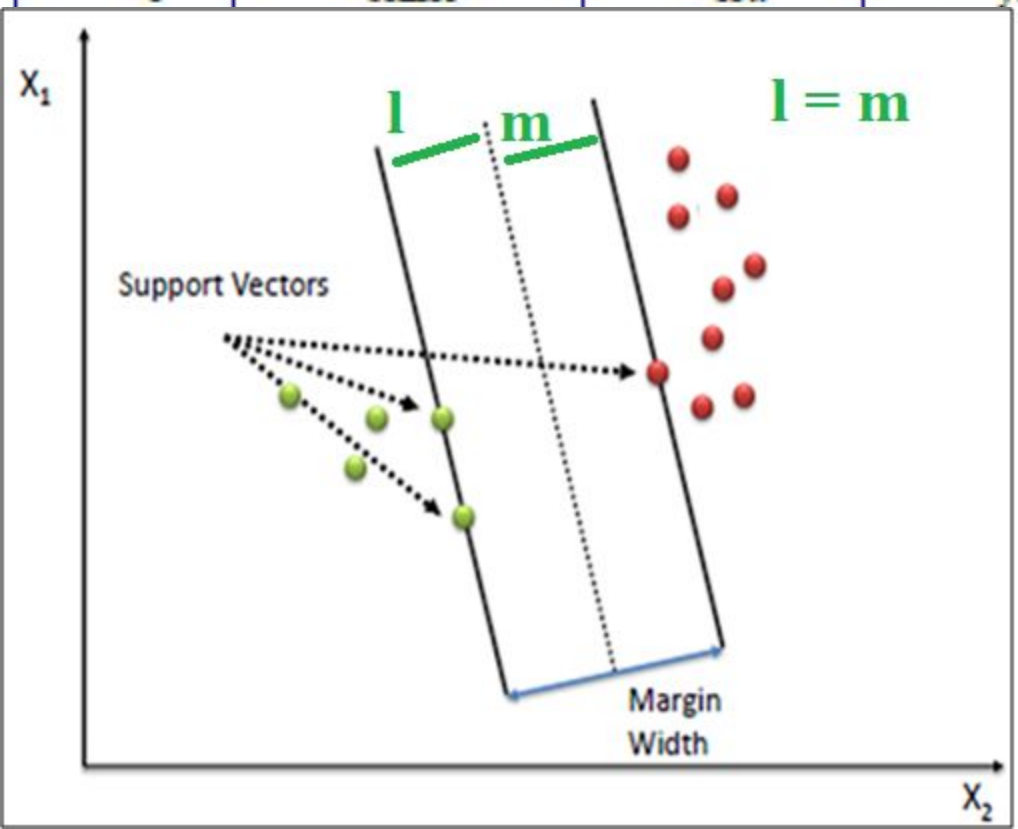
What is a Support Vector Machine?

- Support vector machines (SVMs), a method for the classification of both linear and nonlinear data.
- In a nutshell, an SVM is an algorithm that works as follows. It uses a nonlinear m
higher dimension.
- Within this new d
hyperplane (i.e., a
class from another
- With an appro
dimension, data
hyperplane.
- The SVM finds
training tuples) an



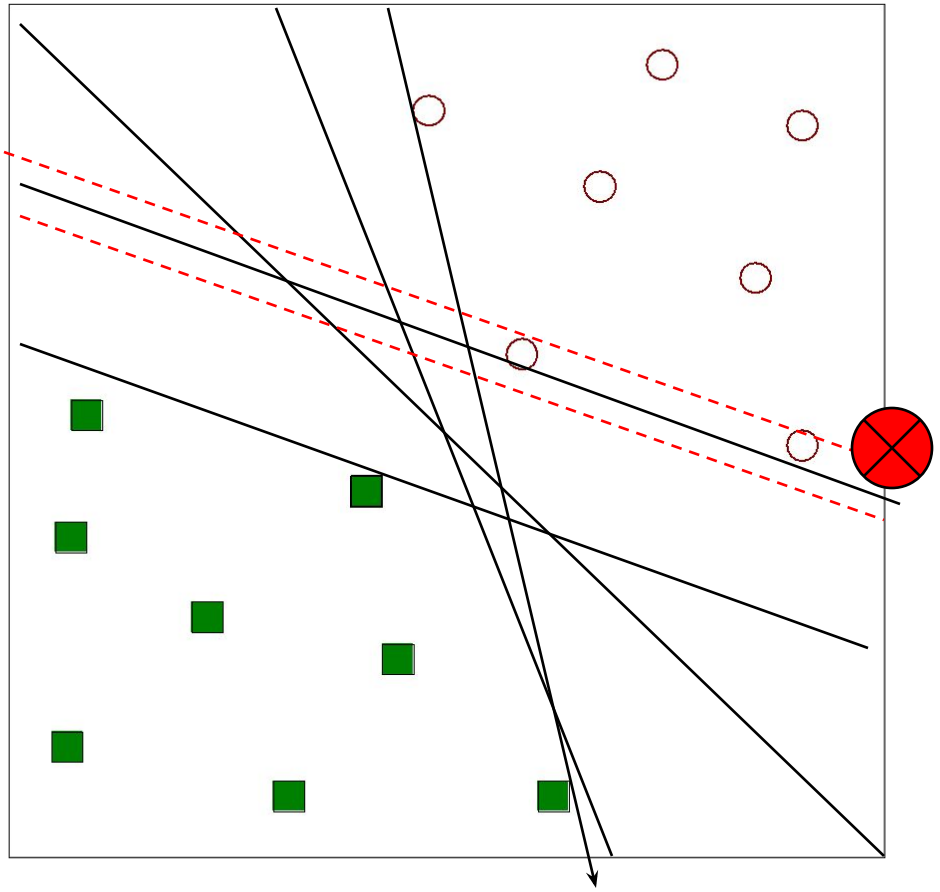
Class-Labeled Training Tuples from the *AllElectronics* Customer Database

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
			es	excellent	no
			es	excellent	yes
			o	fair	no
			es	fair	yes
			es	fair	yes
			es	excellent	yes
			o	excellent	yes
			es	fair	yes
			o	excellent	no



9 buys_computer = yes, 5 = no

Support Vector Machines



- Find a linear hyperplane (decision boundary) that will separate the data

Training set → Hyperplane (candidates) → Decision boundary (optimal hyperplane) → Support vectors (determined by boundary)

.**Training Set (comes first)**

1. You must have labeled training data before anything else.
2. Example: points in 2D space with class labels (+1, -1).

.**Hyperplane (candidate decision boundaries)**

1. The SVM algorithm tries to find a separating hyperplane between the two classes.
2. Initially, many possible hyperplanes exist.

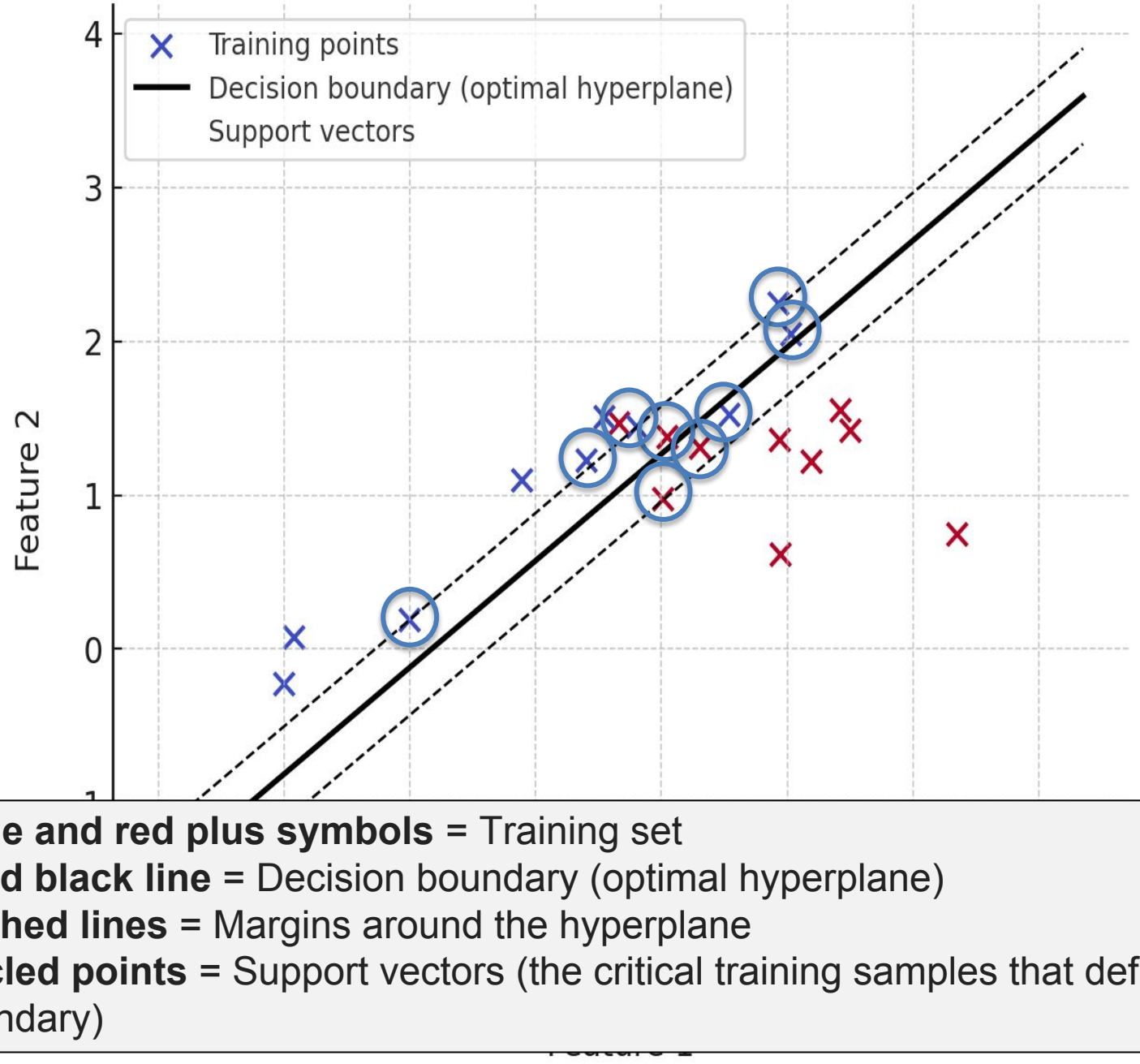
.**Decision Boundary (final chosen hyperplane)**

1. Among all hyperplanes, SVM selects the one with the **maximum margin** (the widest separation between classes).
2. This final separating line (in 2D) or plane (in higher dimensions) is the **decision boundary**.

.**Support Vectors (identified last)**

1. After the optimal hyperplane is chosen, the **support vectors** are the training points lying closest to the boundary.
2. They are the critical points that define and "support" the decision boundary.

SVM: Training set → Hyperplane → Decision boundary → Support vectors



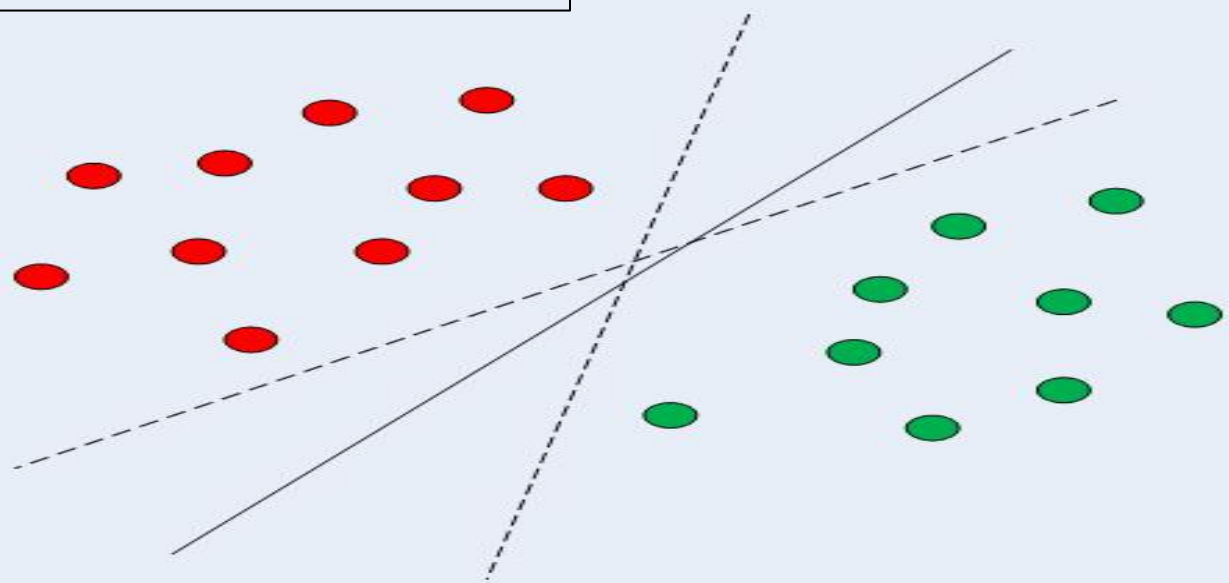
- **Blue and red plus symbols** = Training set
- **Solid black line** = Decision boundary (optimal hyperplane)
- **Dashed lines** = Margins around the hyperplane
- **Circled points** = Support vectors (the critical training samples that define the boundary)

Support Vector Machines

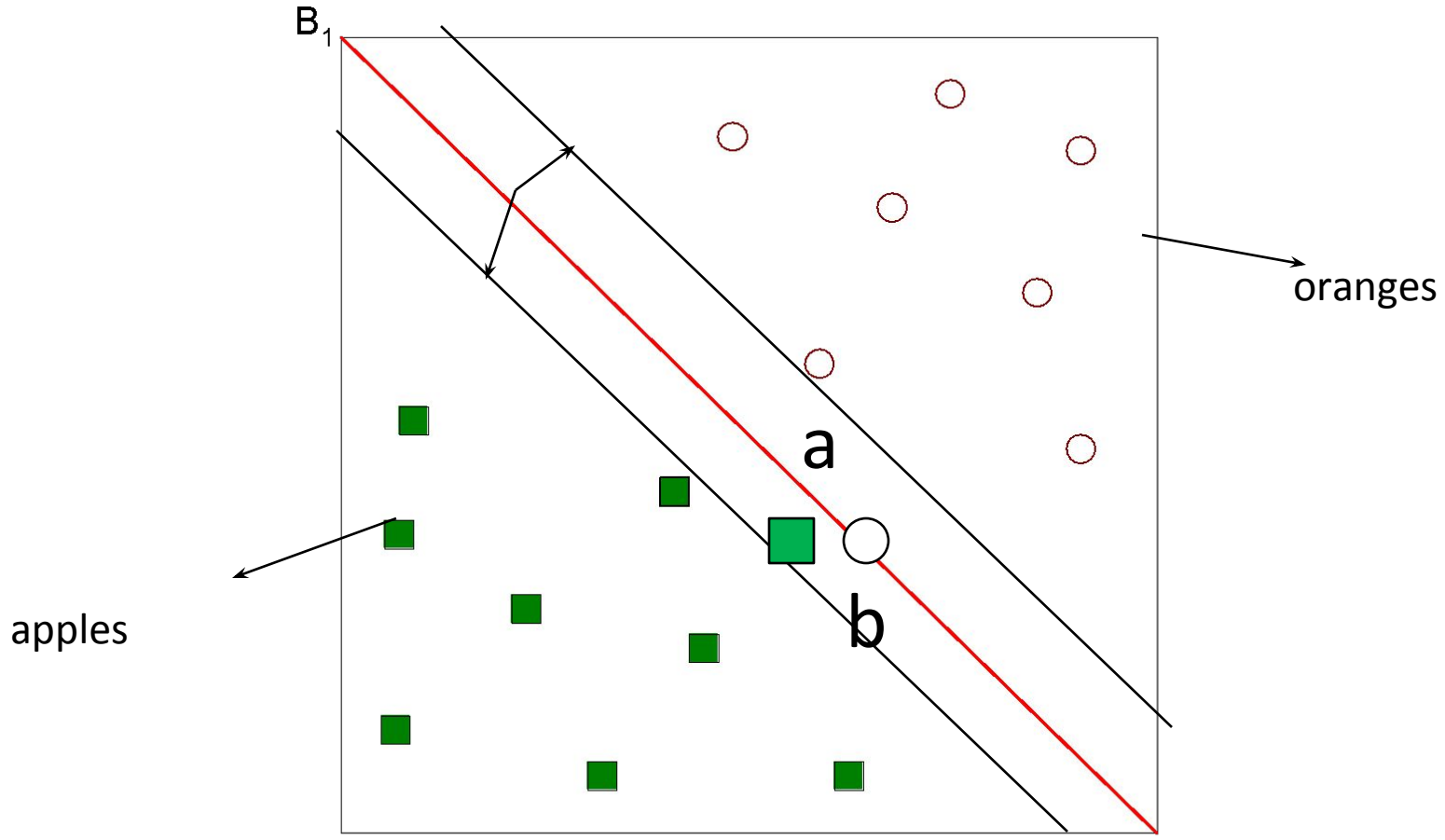
- In case of linearly separable data in two dimensions, as shown in Fig. 1, a typical machine learning algorithm tries to find a boundary that divides the data in such a way that the misclassification error can be minimized.
- If we closely look at Fig. 1, there can be several boundaries that correctly divide the data points.
- The two dashed lines as well as one solid line classify the data correctly.

Multiple Decision Boundaries

Fig. 1

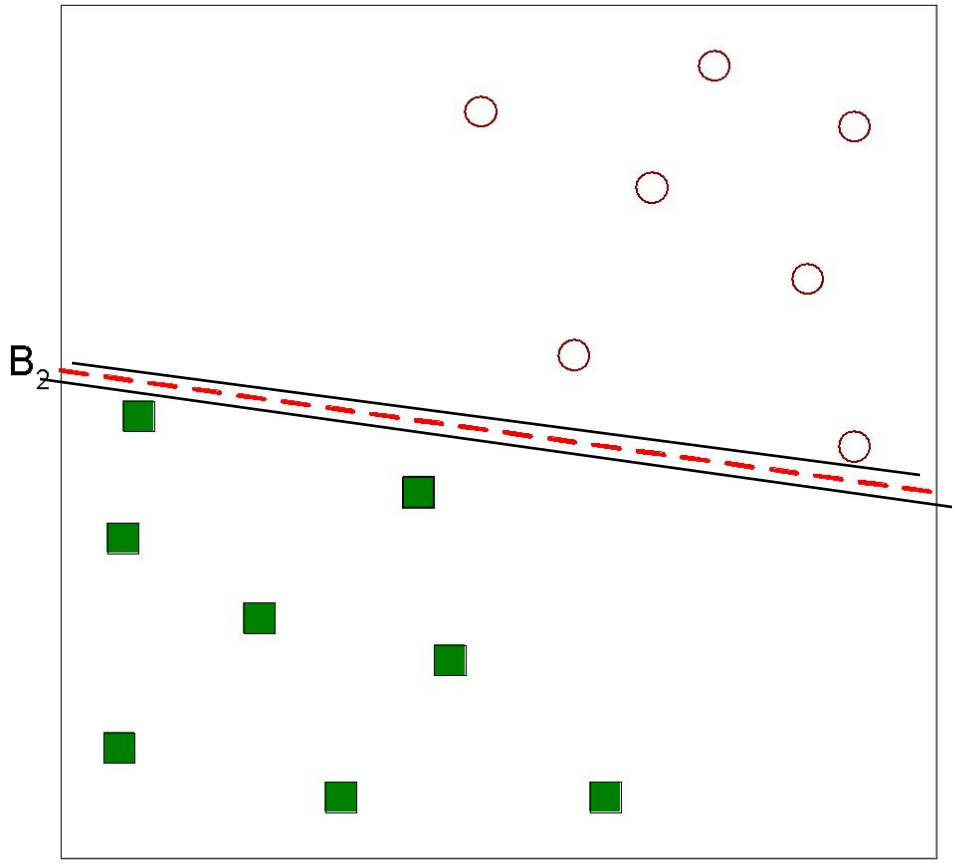


Support Vector Machines



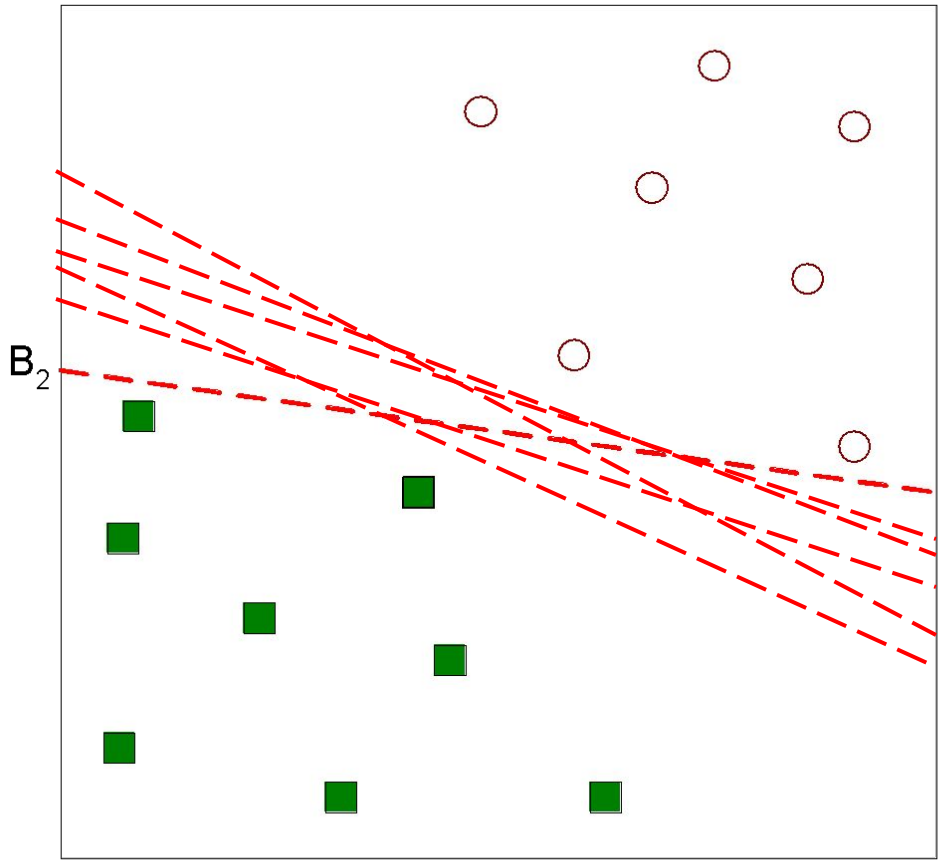
- One Possible Solution

Support Vector Machines



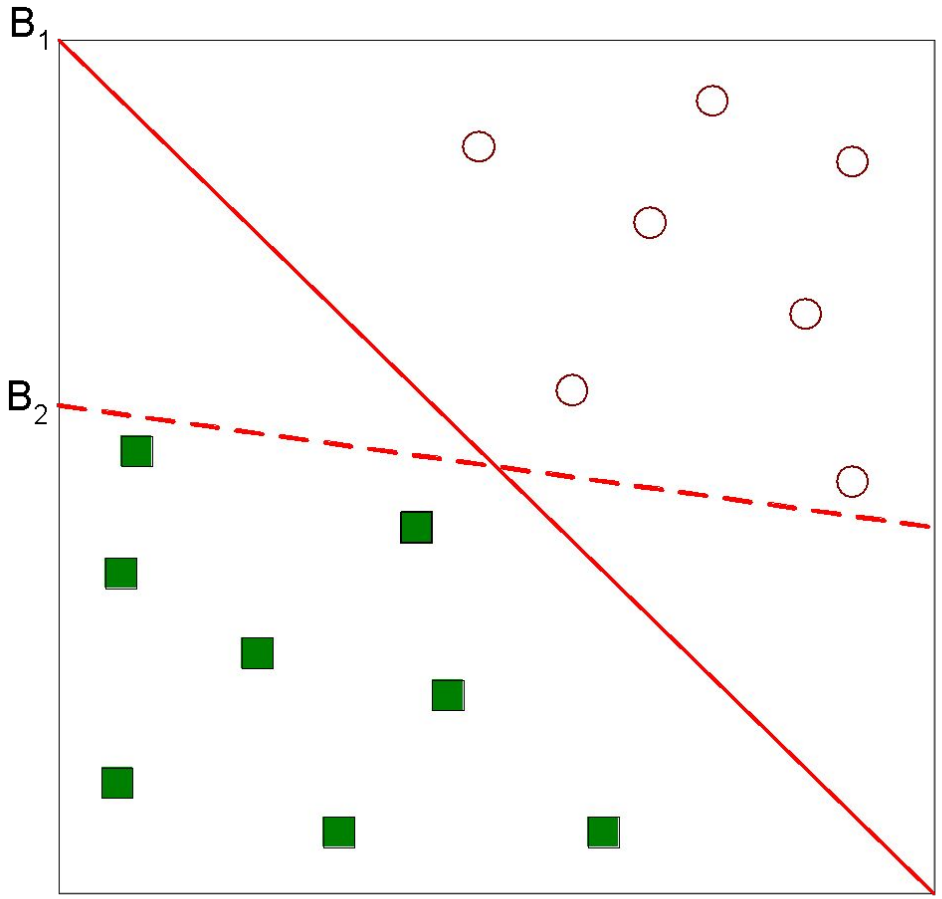
- Another possible solution

Support Vector Machines



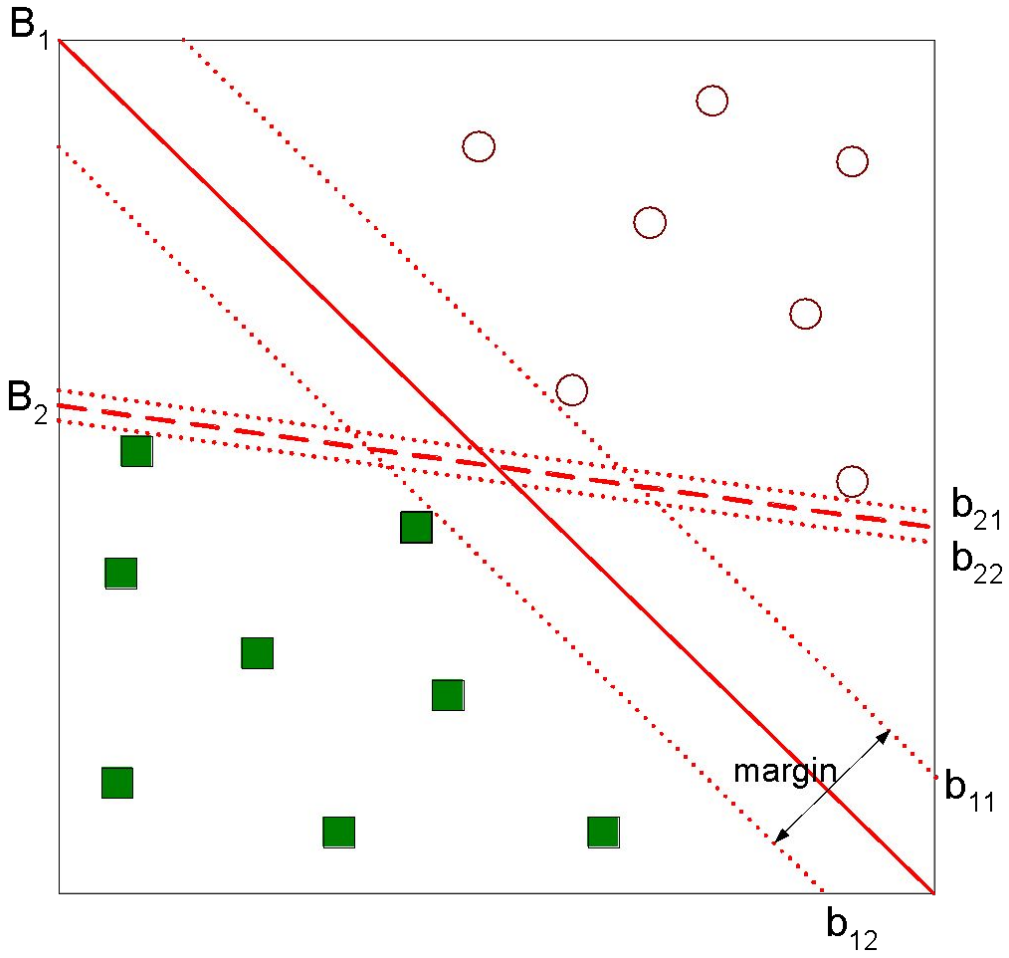
- Other possible solutions

Support Vector Machines



- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines



- Find hyperplane maximizes the margin => B1 is better than B2

SVM differs from the other classification algorithms in the way that it chooses the *decision boundary* that maximizes the distance from the nearest data points of all the classes.

An SVM finds the *most optimal decision boundary*.

The most optimal decision boundary is the one which has maximum margin from the nearest points of all the classes. The nearest points from the decision boundary that maximize the distance between the decision boundary and the points are called support vectors as seen in Fig 2. The decision boundary in case of support vector machines is called the *maximum margin classifier, or the maximum margin hyper plane*.

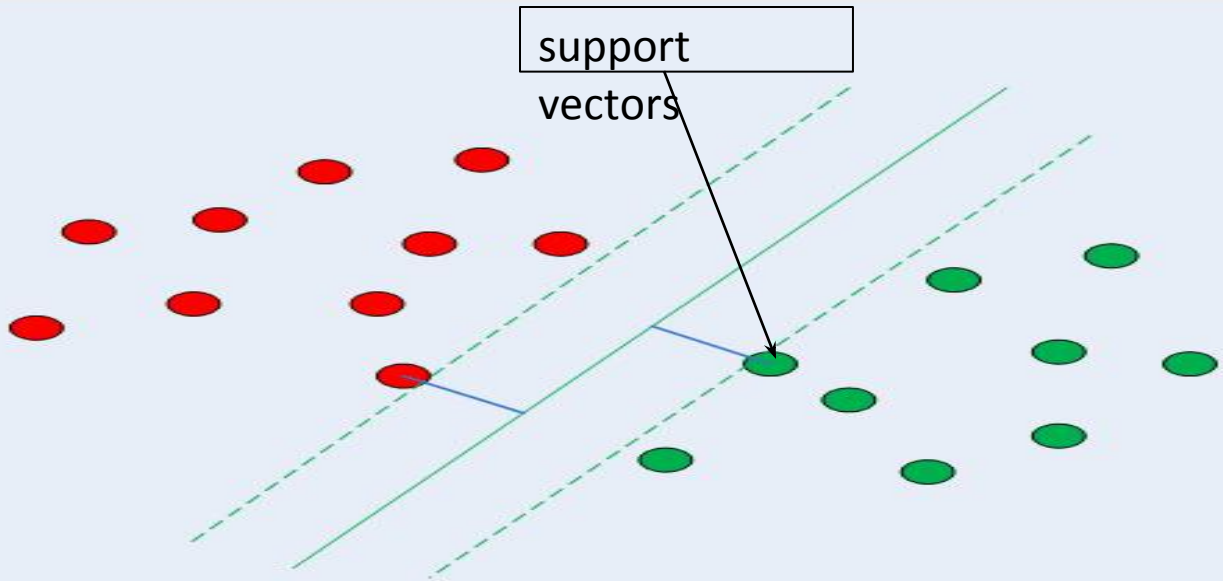
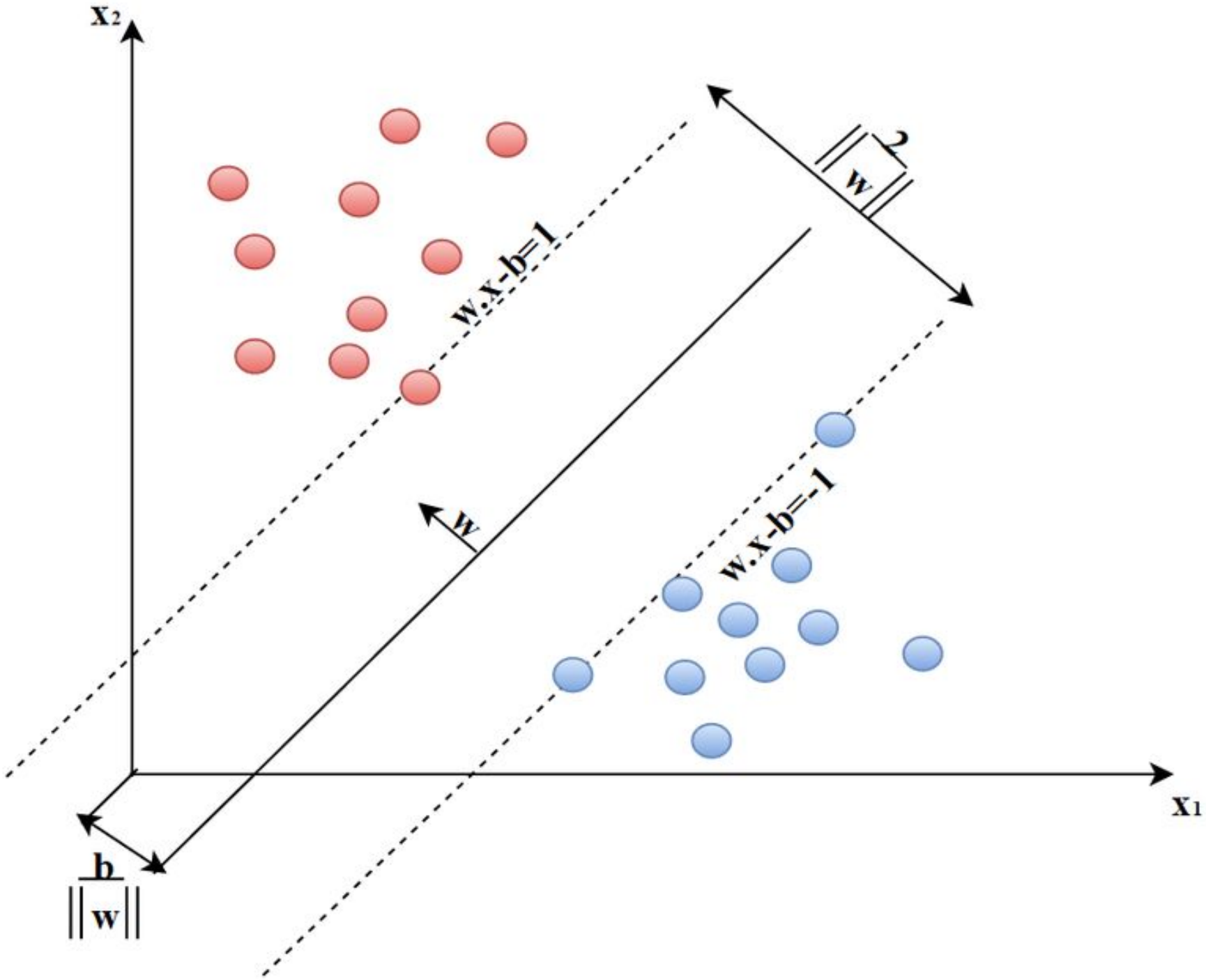
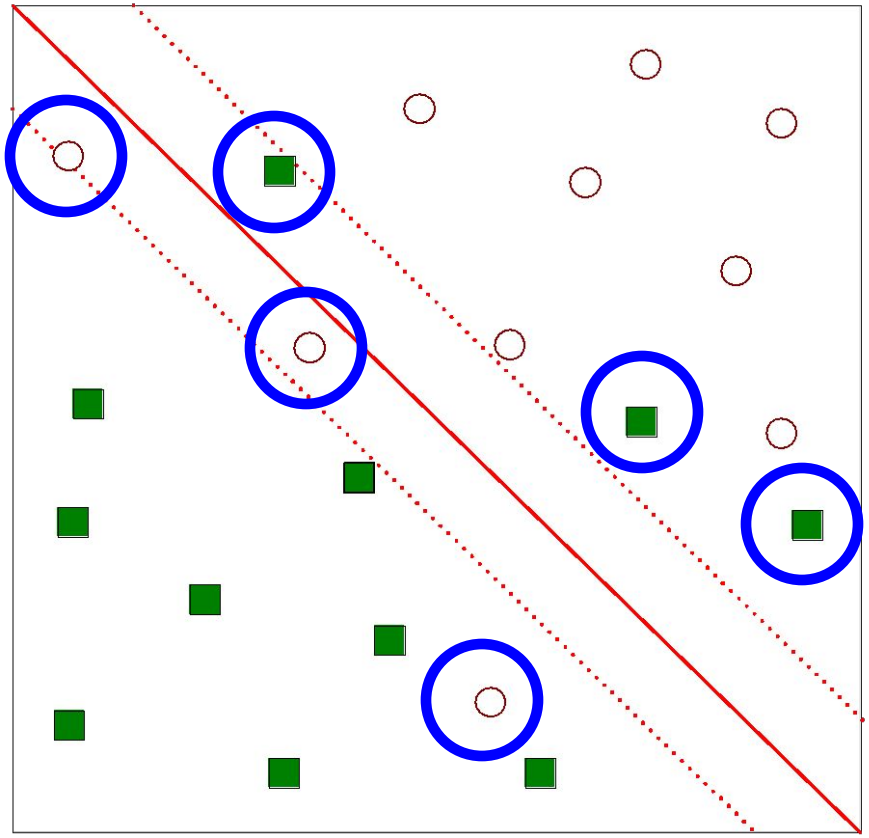


Fig 2

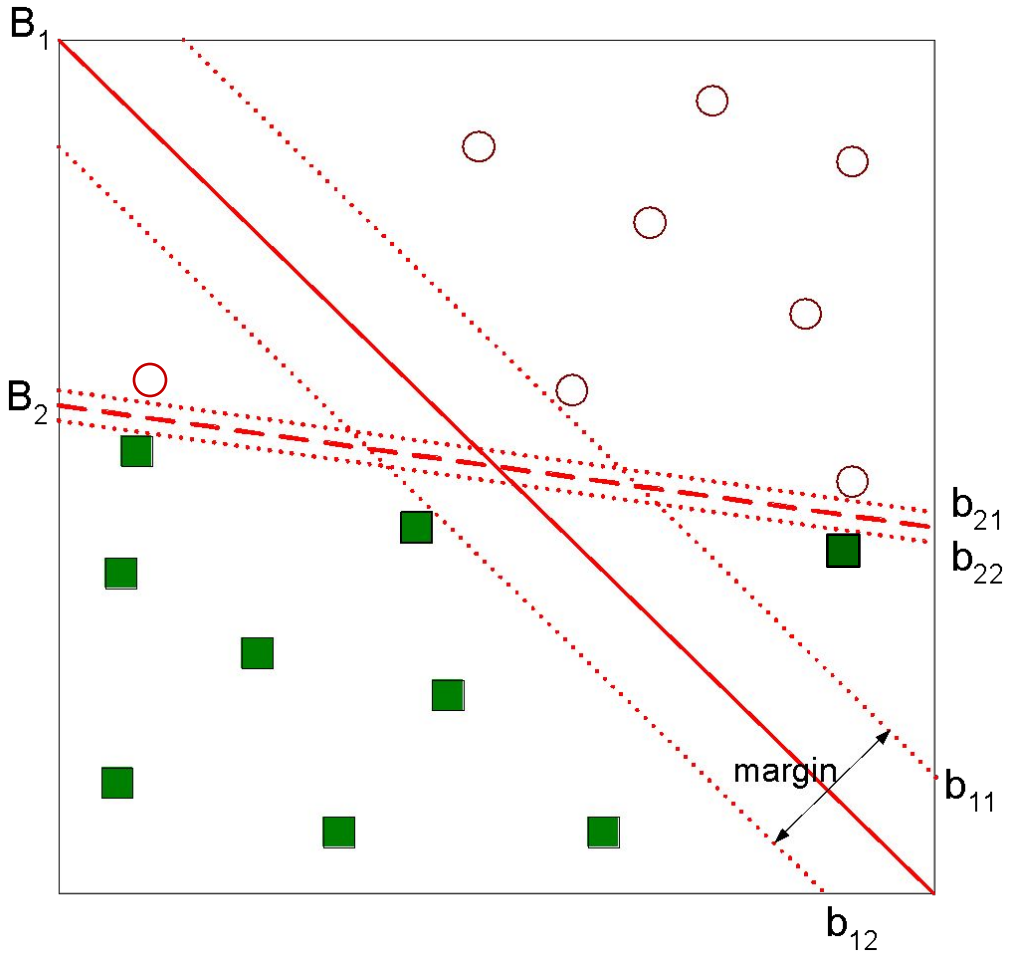

x2
x1
w.x-b=1
w.x-b=-1
w
2
w
b
||w||

Support Vector Machines

- What if the problem is not linearly separable?



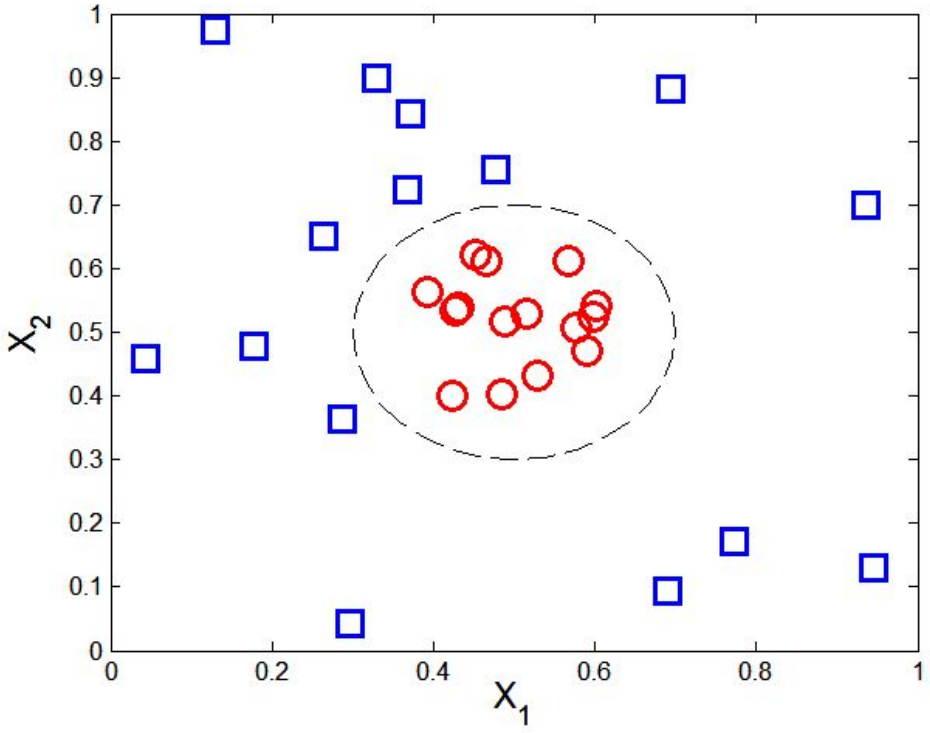
Support Vector Machines



- Find the hyperplane that optimizes both factors

Nonlinear Support Vector Machines

- What if decision boundary is not linear?



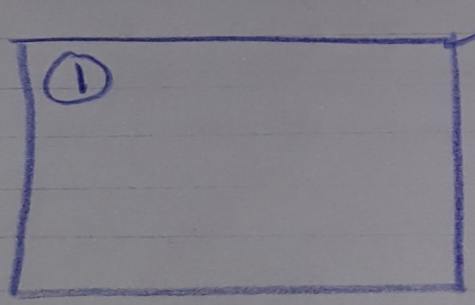
$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

Support Vector Machine (SVM)

Applications where we use SVM :- 1) Handwritten character or digit recognition.
2) Text categorization, 3) High dimensional Data → SVM solves the problem of high dimensionality

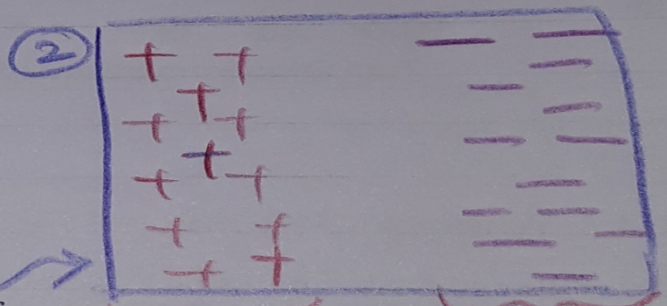
There are two Terms / Words in SVM :- Support Vector
Vectors in machine learning, in our case data mining are data or training examples / training instances, i.e, with help of training examples we construct machines or classifiers.

Support vector :- In SVM, we use the subset of training data and it is used to represent decision boundary



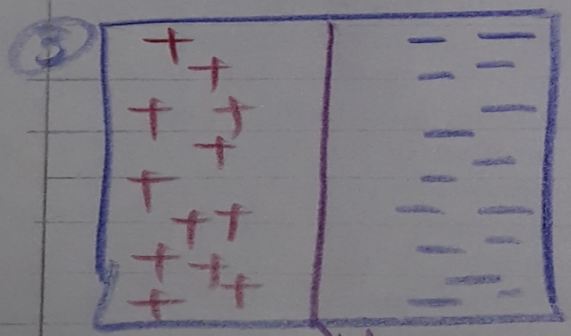
decision boundary

We are going to plot some random dataset into decision boundary.

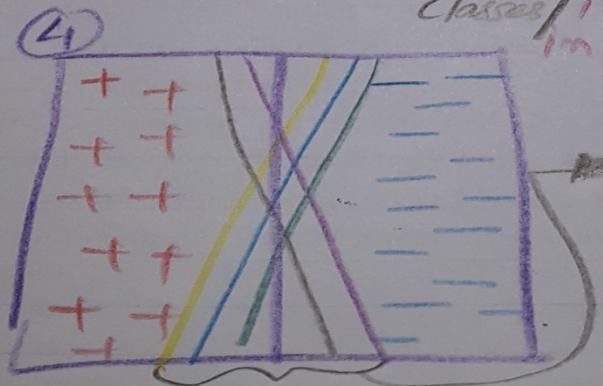


positive classes / instances in LHS

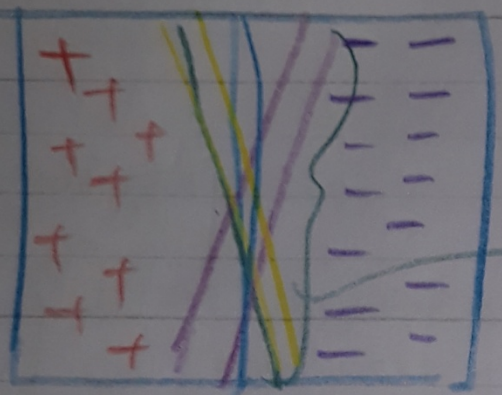
negative classes or instances in RHS



decision boundary



All these lines are hyperplanes used to separate the dataset for being classified as positive or negative instances.



1) One important aspect of SVM to remember is: SVM works on the concept of maximal margin Hyperplanes.

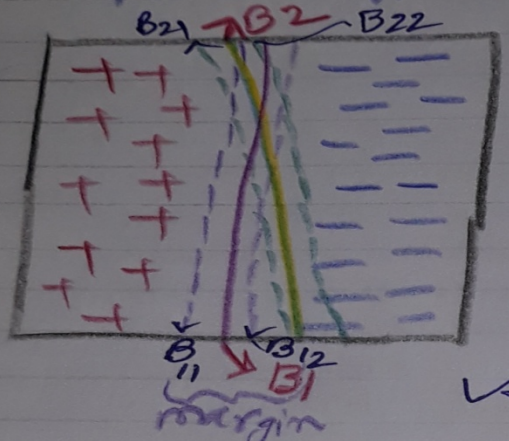
→ These are hyperplanes

We have consider some attributes of hyperplanes

1) Infinite in number
2) Choose any of these hyperplanes that is the most optimal.
3) Training errors on these hyperplanes are Zero, i.e., training err = 0
4) We have to choose any one of these hyperplanes, the case can happen that the chosen hyperplane will not perform well in future with unseen data set/example. equally

i.e, major drawback of hyperplanes is: Today we have selected one hyperplane, tomorrow it may not equally perform on some different training examples, i.e, we can not generalize the performance of hyperplanes, We have to be careful when choosing hyperplanes for Linear SVM.

How Hyperplanes perform in Linear SVM?



How hyperplane affects the performance of Linear SVM?

✓ B1 = boundary B1; B2 = boundary B2

✓ There is an area spanned by these boundaries, say, area owned by B1 and area owned by B2. We have labeled these areas as {B11, B12} for B1, {B21, B22} for B2.

✓ The gaps between two boundaries are called margin.

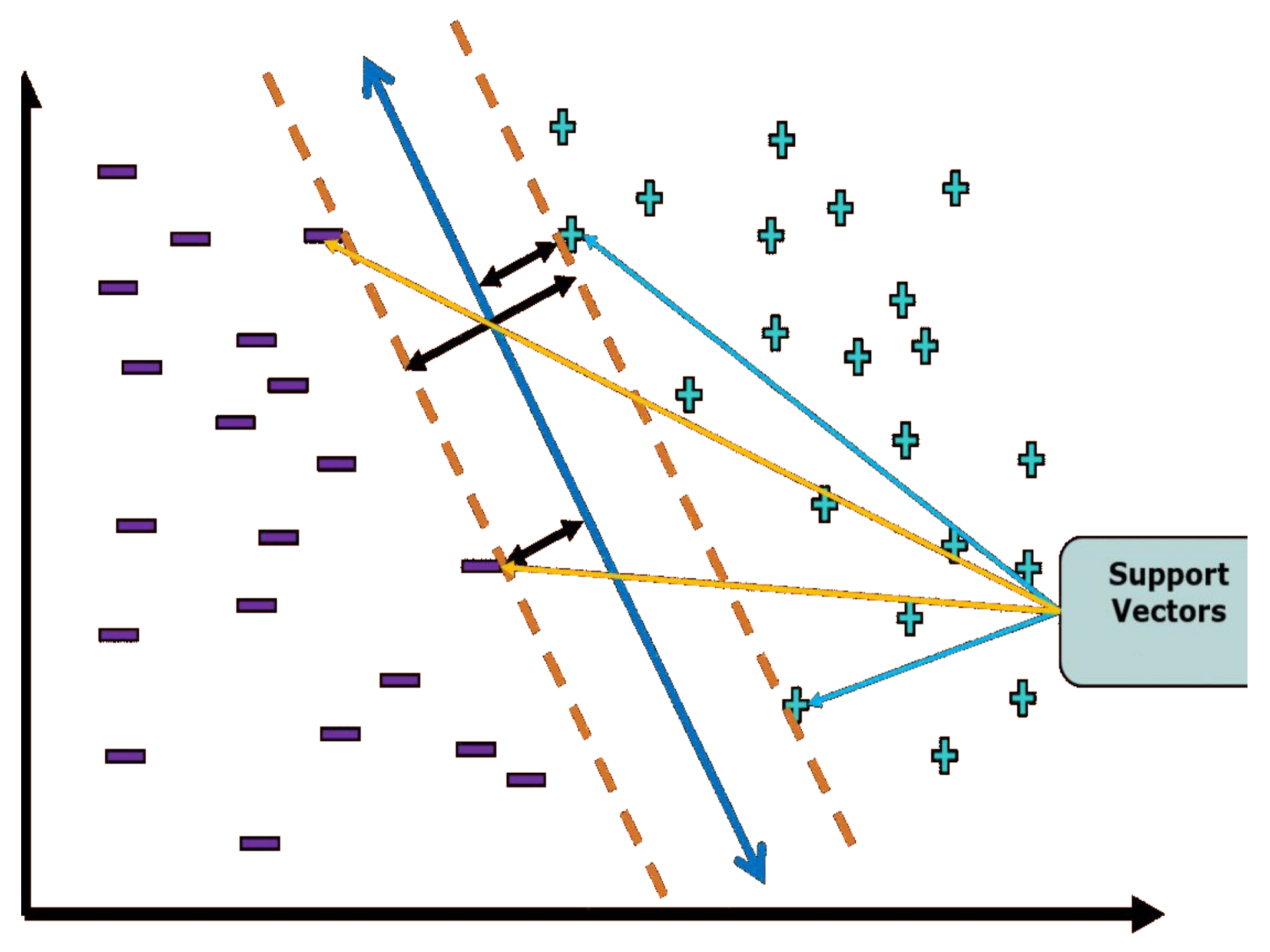
✓ If your hyperplane's margin is higher, then go with the selected hyperplane. Here we have margin of B1 or is greater than that of B2, i.e., B1 > B2 i.e., we will consider B1 and we will go for further classification. This means, we are talking about Marginal Hyperplanes, i.e., maximal margin hyperplanes.

What is the reason for choosing highest margin hyperplanes?

(Higher) margin ↑ → better control over generalization error.

(Lesser) margin ↓ → the model may go through overfitting.

This discussion on hyperplane, margin, maximal margin has link with statistical approach known as SRM, i.e., Structural Risk Minimization.


Support
Vectors

Hyperplanes to Structural Risk Minimization

For any classifier, we have $R \leq R_e + \Phi\left(\frac{h}{N}, \frac{\log \delta}{N}\right)$

- R: true error
- R_e: training error
- Φ: monotone increasing function
- δ: eta / probability
- h: capacity of how many hyperplanes

Once again the formula :-

$$R \leq R_e + \Phi\left(\frac{h}{N}, \frac{\log \delta}{N}\right)$$

There is a relation between the capacity (h) and margin

$$h \propto \frac{1}{\text{margin}}$$

1) If you have higher capacity your margin should be small

$\downarrow$ margin $\longrightarrow$ $h \uparrow$

$\uparrow$ margin $\longrightarrow$ $h \downarrow$

We will go for $\{\uparrow$ margin and $\downarrow h\} \Rightarrow$ Talking about maximal margin hyperplanes or Linear SVMs or Classifier

Maximal Margin Classifiers (Linear SVM) [or hyperplanes]

Why? It selects that hyperplane that has the highest margin (largest margin) among all the hyperplanes.

✓ Let as consider a binary classification problem.

✓ Let's take two variables, x_i and y_i and represent as in a tuple format → (x_i, y_i)

Where x_i can take values like $x_i \Rightarrow 1, 2, 3, \ldots N$ where N is the total number of training instances.

So, we have binary classification problem for N training examples and we have y_i, i,e, the class labels, i,e, $y_i \in \{-1, +1\}$ (Binary classification)

Equation for a decision boundary

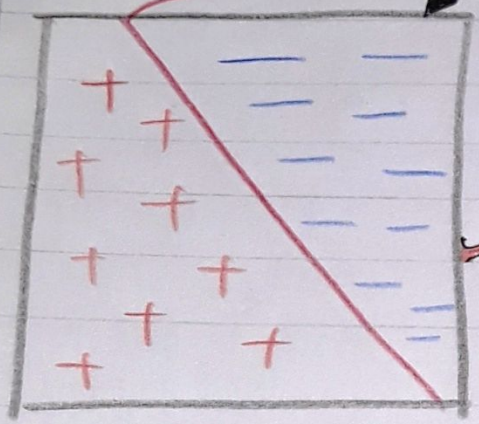
for the case of Linear SVM, we will take the equation in Slope Intercept form: $y = m * x + c$ ⇒ midpoint

Let's replace m and c by w and b, where, w is the weight, $w.x + b = 0$

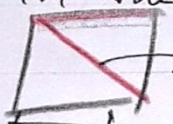
x → Input vector

Maximal Margin Hyperplanes.

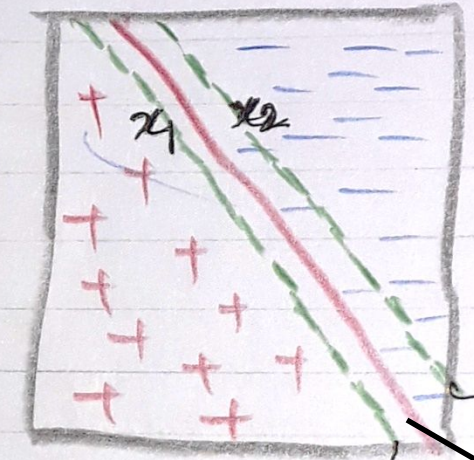
We are going to $w \cdot x + b = 0$ with a decision boundary :- a line in the decision boundary this line ($wx+b=0$) bisects two examples (i.e., training instances with positive (+) and negative (−) class values)



→ We have sets of '+' instances in the left hand side and set of '−' instances in the right hand side of the line.



✓ Now we are going to draw two neighbouring areas of the line

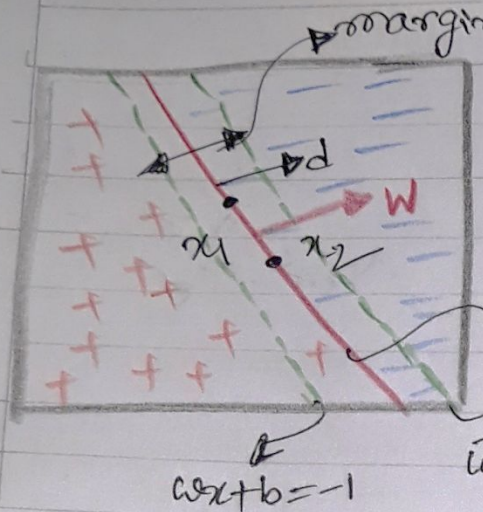


✓ x_1 and x_2 are two portions, one on the L.H.S, and the other on R.H.S.

✓ Equation for the dashed line below the solid line is :- $w \cdot x + b = -1$ and equation for the line above the solid line is :- $wx+b=+1$

wx + b = 0

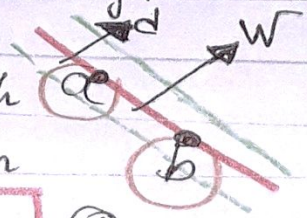
Maximal Margin Hyperplanes



✓ Margin :- Gap between ($wx+b=-1$) and ($wx+b=+1$) is called margin, i,e, the area swept across the left & right of hyperplane.

✓ margin is represented as (d).

✓ We also have a weight vector, (W), that is perpendicular to this decision boundary.



✓ Let's write some equation with two points, (a) and (b) on the main hyperplane :-

$$w \cdot x_a + b = 0 \quad \text{—(1)}$$
$$w \cdot x_b + b = 0 \quad \text{—(2)}$$

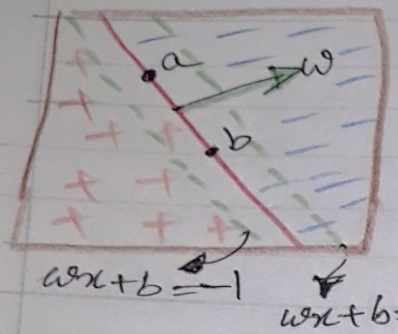
If we try to subtract (1) − (2) :- $w \cdot (x_a - x_b) = 0$

→ Vector parallel to our decision boundary (DB)

→ $w \perp DB$ (W is perpendicular to the DB)

Let's move on to make some generalised statement, where we put some instance and try to predict the class $\{-1, +1\}$. (i.e., Classify test instance)

How to classify test data with Maximal Margin Hyperplane



- Let's have a variable k
- We have a equation for the line above the hyperplane, i.e, in the area with negative instances, $w \cdot x_- + b = 0$

area below the hyperplane $w \cdot x_+ + b = 0$

- Now we have the variable k, so, let's utilize it,

$w \cdot x_- + b$, $k > 0$, area above the hyperplane

$w \cdot x_+ + b$, $k < 0$, area below the hyperplane

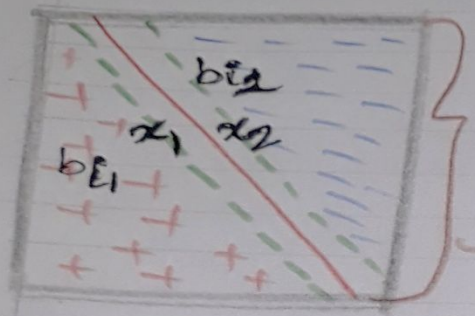
- Now, let's construct some equation that fits the model, say, we are working a training instance, z.
- Say we have two values for binary classification problem,

$$y = \begin{cases} +1, \text{if } w \cdot z + b > 0 \\ -1, \text{if } w \cdot z + b < 0 \end{cases}$$

For any training example, we are trying to see if we can fit in either positive or negative class. i,e, make prediction of which particular class the test example will fall in or fit.

How to Calculate the margin of a linear SVM

Calculate the Margin of a Linear SVM



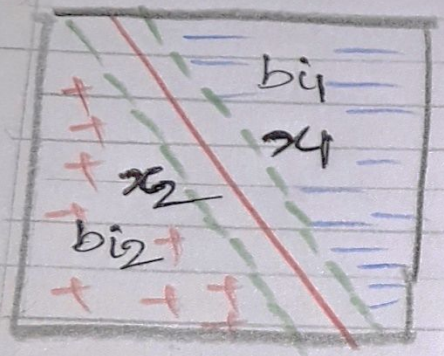
We are going to construct two equations :-

bi_2 → upper bound, taking the boundary of the hyperplane; bi_1 → lower bound

$bi_2 : w.x_2 + b = 1$ ---- (1)

$bi_1 : w.x_1 + b = -1$ ---- (2)

If we subtract (1) from (2), (making x_1 coming first) then we get, $w(x_1 - x_2) = (-2)$ → if we want to make it +positive, then, we have to change the bounds,



$bi_1 : w.x_1 + b = 1$ ---- (1)

$bi_2 : w.x_2 + b = -1$ ---- (2)

w, b ⇒ model parameters

↳ Intrinsic to the classifier

making $bi_1 - bi_2 \Rightarrow$ $w(x_1 - x_2) = 2$ (*)

(*) → This equation reduces to $||w|| * d = 2$

↳ Vector norm

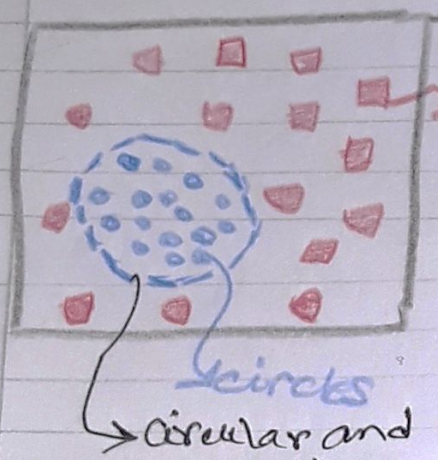
we know $(x_1 - x_2) = d$ = margin

i.e, $d = \frac{2}{||w||}$

Final value of a margin in Linear SVM or maximal margin classifier

NonLinear SVM

✓ Hyperplane that we have Learned in Maximal Margin Hyperplanes bisects the training instances.



✓ In order to classify squares from circles, we have to draw a decision boundary around circles.

✓ The decision boundary created is of circular nature and NOT of Linear type.

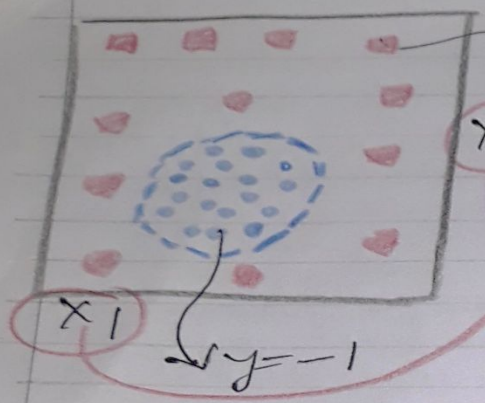
✓ In this dataset, all circles are clustered together around the center and the squares are distributed all throughout the dataset (i.e, a 2D plane).

✓ We have to transform this dataset to get a decision boundary changing from a circular (to a noncircular) OR a Linear dataset / Linear decision boundary

* How to perform this transformation ?

A tuple vector is represented as (x_i, y_i) where we denote x_i as attribute set, and y_i is the class, represented as $y = \{-1, +1\}$ // binary classification.

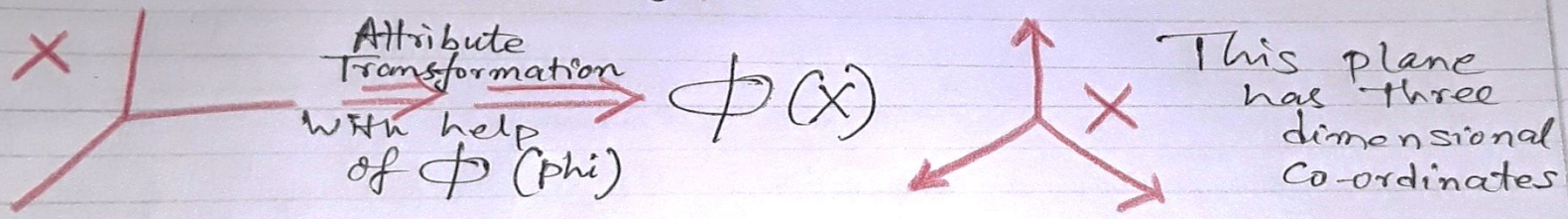
Non Linear SVM



(x_i, y_i) → $y = +1, -1$; $y \Rightarrow$ Class instances

We can not draw a Three dimensional decision boundary, so, I am giving the Circular boundary as X_1 plane & X_2 plane.

Question:- How do we transform this circular decision boundary to a linear decision boundary?



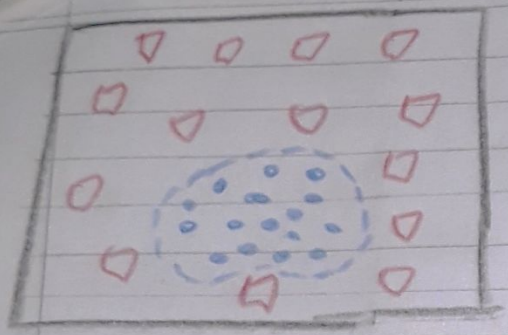
Question:- How to make this attribute transformation (with a goal to convert a circular plane to Linear plane)?

We need to take help of some functions,
In order to classify the training dataset into positive (+1) and negative (-1) instances, we use equations like :-

$$y_{(x_1, x_2)} = \begin{cases} +1, & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 \end{cases}$$

↳ Eucledian distance (Calculate to separate +positive & −negative instances)

Nonlinear SVM



$$y = \begin{cases} +1 & \text{if } \sqrt{(x_1-0.5)^2+(x_2-0.5)^2} > 0.2 \\ -1 & 0.w \end{cases}$$

(x_1, x_2)

We will take the positive side, i.e, $y = +1$ and the equation as follows:-

$$\sqrt{(x_1-0.5)^2+(x_2-0.5)^2} = 0.2$$ // taking equality

Taking squares of both sides ⇒ Converting to a quadratic equation as follows:-

$$x_1^2 - x_1 + x_2^2 - x_2 = 0.46$$ // quadratic equation obtained in the process of transformation

Q:- What is our transformation function ?

We denote as $\phi : (x_1, x_2)$ // we want to transform over variables, x_1 & x_2

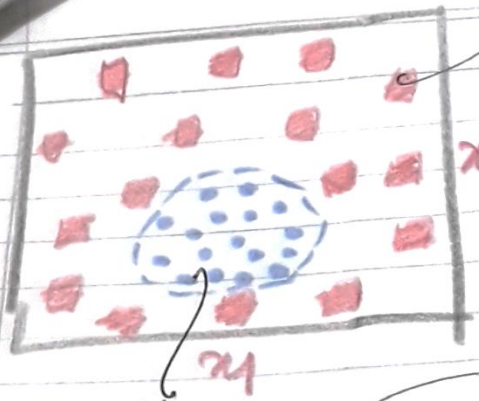
If we transform the above function with two variables, then, we will get the following:-

$$(x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$ // five variables in this non-linear transformation when we go for attribute transformation.

We can also have weights,

Suppose we have five weights, $W = w_0 .. w_4$ //

Non Linear Transformation



Attribute transformation

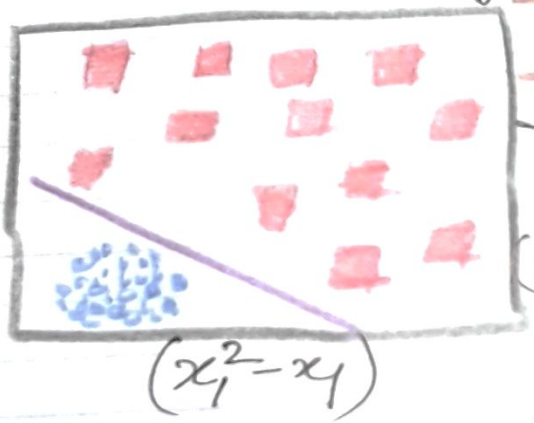
$\phi(x_1, x_2) \Rightarrow x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1$

$W = w_0 \dots w_4$ (// five weights)

Associate five weights with transformed variables

$w_4 \cdot x_1^2 + w_3 \cdot x_2^2 + w_2\sqrt{2}x_1 + w_1 \cdot \sqrt{2}x_2 + w_0 = 0$

This transformation will give us a new decision boundary as :-



$wx + b = 0$

This is a linear decision boundary

We have some trick in non-linear SVM : KERNEL trick

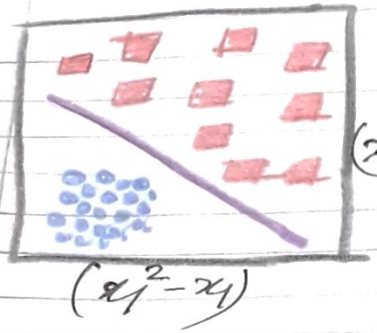
Suppose we have two input vectors, Take their dot product & denote them as similar

$\phi(x_1) \cdot \phi(x_2)$ ($\Rightarrow$ Similar)

or $\phi(u) \cdot \phi(v)$

KERNEL TRICK

$\phi(x_1) \cdot \phi(x_2) \Rightarrow \phi(u) \cdot \phi(v)$



✓ Replace x_1 and x_2 with u and v.

✓ We expand the above dot product :-

$u :- u_1^2, u_2^2, \sqrt{2}u_1, \sqrt{2}u_2, 1$ — (1) // for u, we have u_1 & u_2

$v :- v_1^2, v_2^2, \sqrt{2}v_1, \sqrt{2}v_2, 1$ — (2) // v_1, v_2

We need to make product version of (1) and (2) as follows :-

$u_1^2 v_1^2 + u_2^2 v_2^2 + 2u_1 v_1 + 2u_2 v_2 + 1$

$(u.v + 1)^2$ → Transformation Result

What is a kernel function?

$K(u, v) = \phi(u) \cdot \phi(v)$ → transformation over u & v

$= (u.v + 1)^2$

Kernel function satisfies principle of mathematics → MERCERS theorem

✓ Advantage over attribute transformation : Kernel function is cheaper (in time)

✓ fast in Computation

✓ transformation takes place in original space, there is no need for reorientation.

✓ Transformed space in kernel is known as reproducing Kernel space or Kernel Hilbert's space

Equation of a hyperplane

In an n-dimensional space, a hyperplane is defined as:

$$w \cdot x + b = 0$$

- $w \rightarrow$ weight vector (normal to the hyperplane)

Margin definition

The **margin** is the perpendicular distance between the hyperplane and the nearest data points (support vectors).

Distance of a point x from the hyperplane:

$$\text{distance}(x) = \frac{|w \cdot x + b|}{\|w\|}$$

Thus, margin =

$$\text{Margin} = \frac{2}{\|w\|}$$

(since distance between two margin boundaries is twice the distance from the hyperplane to a support vector).

4. Maximum margin problem

We want to maximize the margin, i.e. maximize $\frac{2}{\|w\|}$.

Equivalent to minimizing $\|w\|^2$.

So the optimization problem becomes: $\min_{w,b} \frac{1}{2}\|w\|^2$

subject to: $y_i(w \cdot x_i + b) \geq 1 \quad \forall i$

Margin hyperplanes

The two parallel margin boundaries are:

$$w \cdot x + b = +1$$

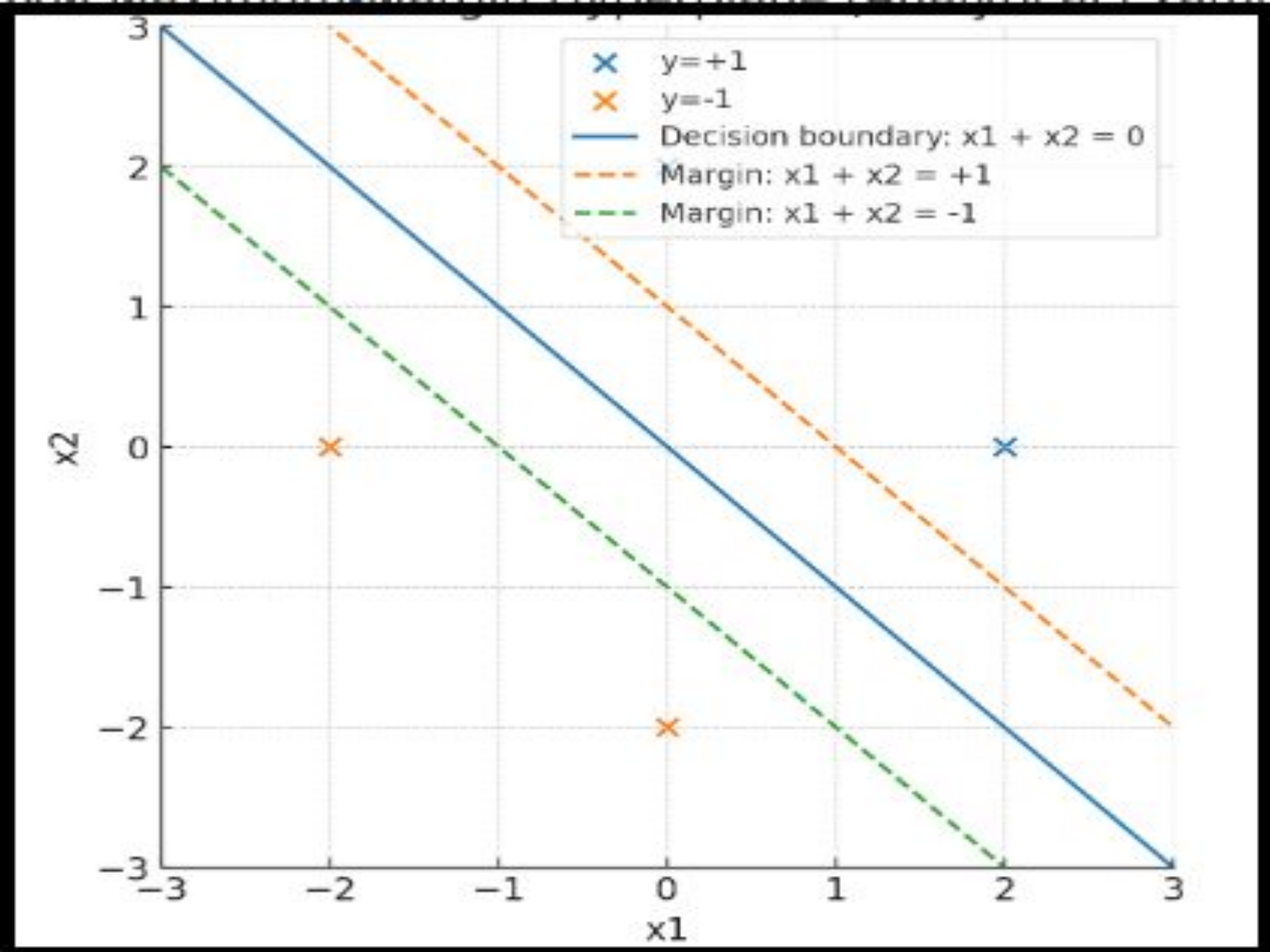
$$w \cdot x + b = -1$$

And the maximum margin hyperplane (decision boundary) is:

$$w \cdot x + b = 0$$

Geometric picture

- Decision boundary: $w \cdot x + b = 0$
- Margins: $w \cdot x + b = \pm 1$
- Support vectors: Points that satisfy $y_i(w \cdot x_i + b) = 1$


y=+1
y=-1
Decision boundary: x1 + x2 = 0
Margin: x1 + x2 = +1
Margin: x1 + x2 = -1
x1
x2
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Dataset

- Positive class (+1): (2, 0), (0, 2)
- Negative class (−1): (−2, 0), (0, −2)

Solve for the maximum-margin separator

We seek a hyperplane $w \cdot x + b = 0$ with constraints

$$y_i \left(w \cdot x_i + b \right) \geq 1.$$

By symmetry, choose $b = 0$ and let $w = (\frac{1}{2}, \frac{1}{2})$. Then:

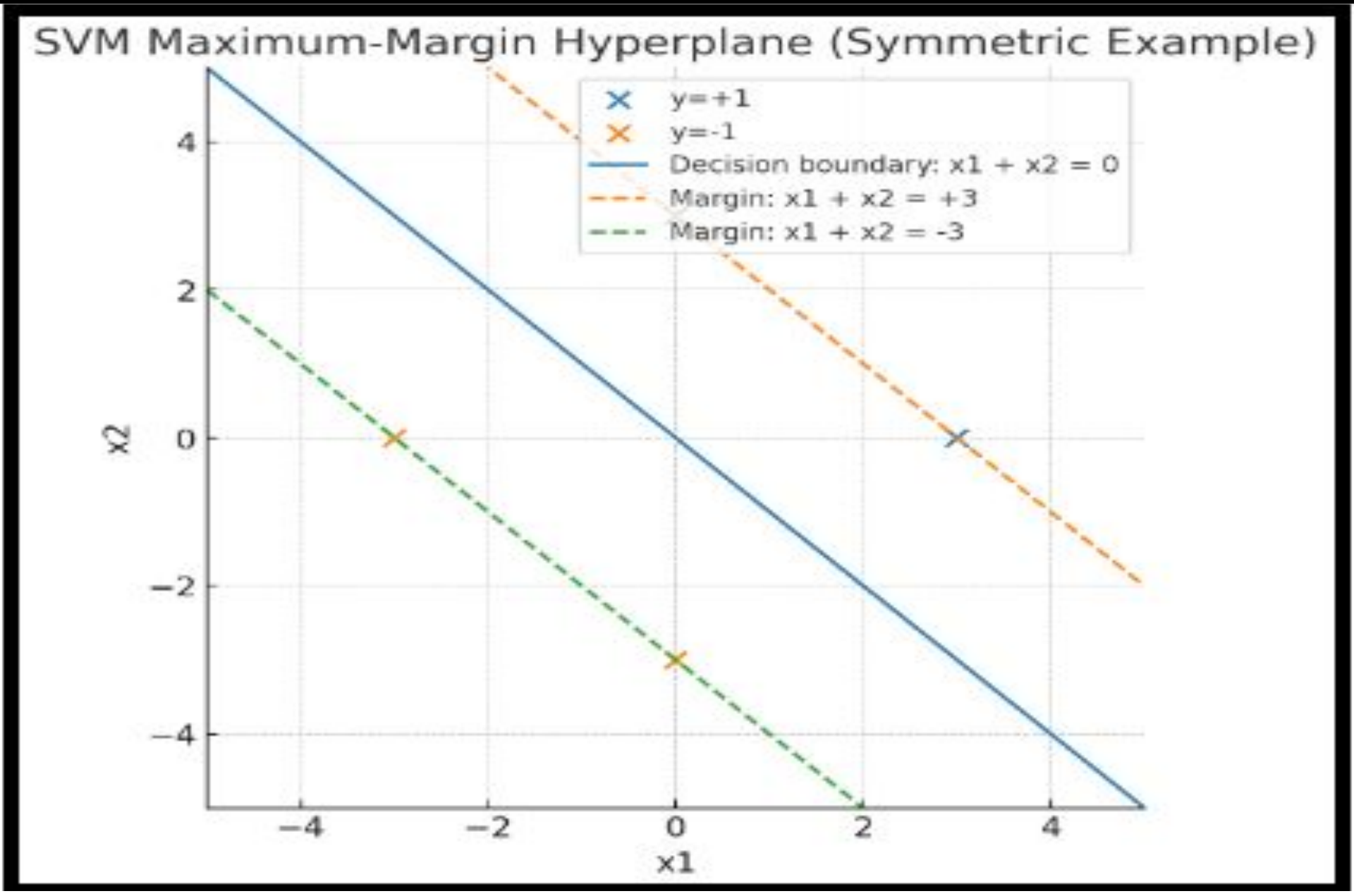
- For $x = (2, 0)$ or $(0, 2)$ with $y = +1$: $w \cdot x = 1 \Rightarrow y(w \cdot x + b) = 1$.
- For $x = (-2, 0)$ or $(0, -2)$ with $y = -1$: $w \cdot x = -1 \Rightarrow y(w \cdot x + b) = 1$.

All four points satisfy the constraint with equality ⇒ they are support vectors.

Final equations

- **Decision boundary (MMH):** $w \cdot x + b = 0 \;\Rightarrow\; x_1 + x_2 = 0$
- **Margins:** $w \cdot x + b = \pm 1 \;\Rightarrow\; x_1 + x_2 = \pm 1$
- **Norm:** $\|w\| = \sqrt{(1/2)^2 + (1/2)^2} = \sqrt{1/2} = 1/\sqrt{2}$
- **Distance (boundary → either margin):** $1/\|w\| = \sqrt{2}$
- **Margin width (between margins):** $2/\|w\| = 2\sqrt{2}$

SVM - Maximal Margin Hyperplane where the classes are symmetric, such as positive points at (3,0) and (0,3), and negative points at (-3,0) and (0,-3).



Data

- $+1$: $(3,0)$, $(0,3)$
- -1: $(-3,0)$, $(0,-3)$

Solve the hard-margin SVM by hand

We want a hyperplane $w \cdot x + b = 0$ with constraints $y_i(w \cdot x_i + b) \geq 1$.

By symmetry, take $b = 0$ and $w = (a, a)$. Enforce the margin on a positive support vector, e.g. $(3,0)$:

$$(3,0): \quad y = +1, \;\; w \cdot x = 3a = 1 \;\Rightarrow\; a = \tfrac{1}{3}.$$

Thus

$$w = \left(\tfrac{1}{3}, \tfrac{1}{3}\right), \quad b = 0.$$

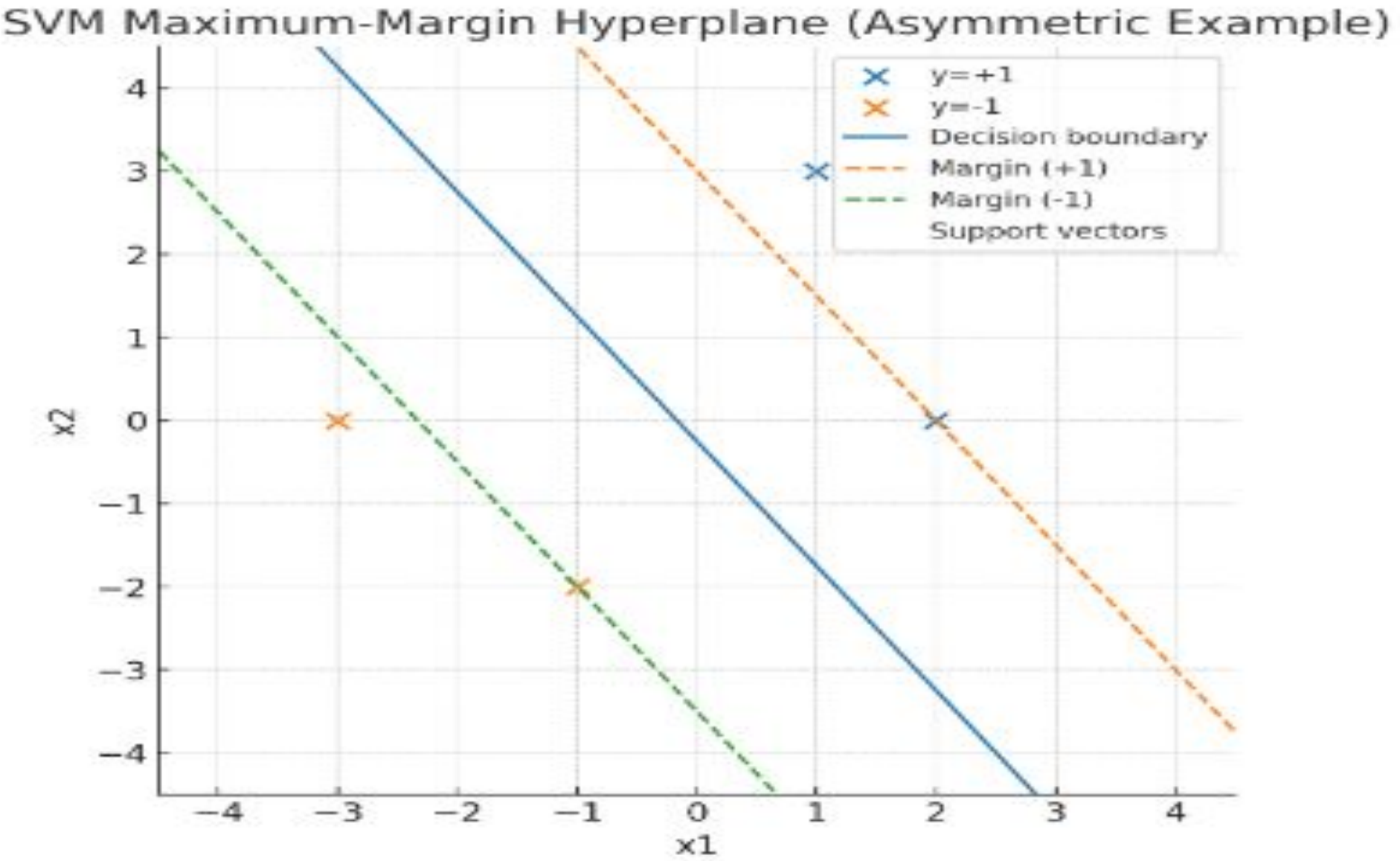
Equations you can draw directly

Margin size

$$\|w\| = \sqrt{\left(\tfrac{1}{3}\right)^2 + \left(\tfrac{1}{3}\right)^2} = \frac{\sqrt{2}}{3}.$$

- Distance (boundary → either margin): $\dfrac{1}{\|w\|} = \dfrac{3}{\sqrt{2}} \approx 2.1213.$
- Margin width (between margins): $\dfrac{2}{\|w\|} = \dfrac{6}{\sqrt{2}} = 3\sqrt{2} \approx 4.2426.$

SVM - Maximal Margin Hyperplane where the classes are asymmetric, such as positive points at (2,0) and (1,3), and negative points at (-3,0) and (-1,-2).



Dataset

- $+1$: $(2,0),\ (1,3)$
- -1: $(-3,0),\ (-1,-2)$

Result (maximum-margin separator)

From the hard-margin SVM (solved exactly for this set):

- Normal vector $w = \left(\frac{6}{13}, \frac{4}{13}\right)$
- Intercept $b = \frac{1}{13}$

Decision boundary (MMH)

$$w \cdot x + b = 0 \iff \frac{6}{13}x_1 + \frac{4}{13}x_2 + \frac{1}{13} = 0 \iff \boxed{6x_1 + 4x_2 + 1 = 0}.$$

Margin lines

$$w \cdot x + b = \pm 1 \iff \boxed{6x_1 + 4x_2 + 1 = \pm 13}.$$

Support vectors

- $(2,0)$: $w \cdot x + b = +1$
- $(-1,-2)$: $w \cdot x + b = -1$

(These are the two support vectors—one from each class.)

Margin size

$$\|w\| = \sqrt{\left(\tfrac{6}{13}\right)^2 + \left(\tfrac{4}{13}\right)^2} = \frac{2}{\sqrt{13}} \approx 0.5547,$$

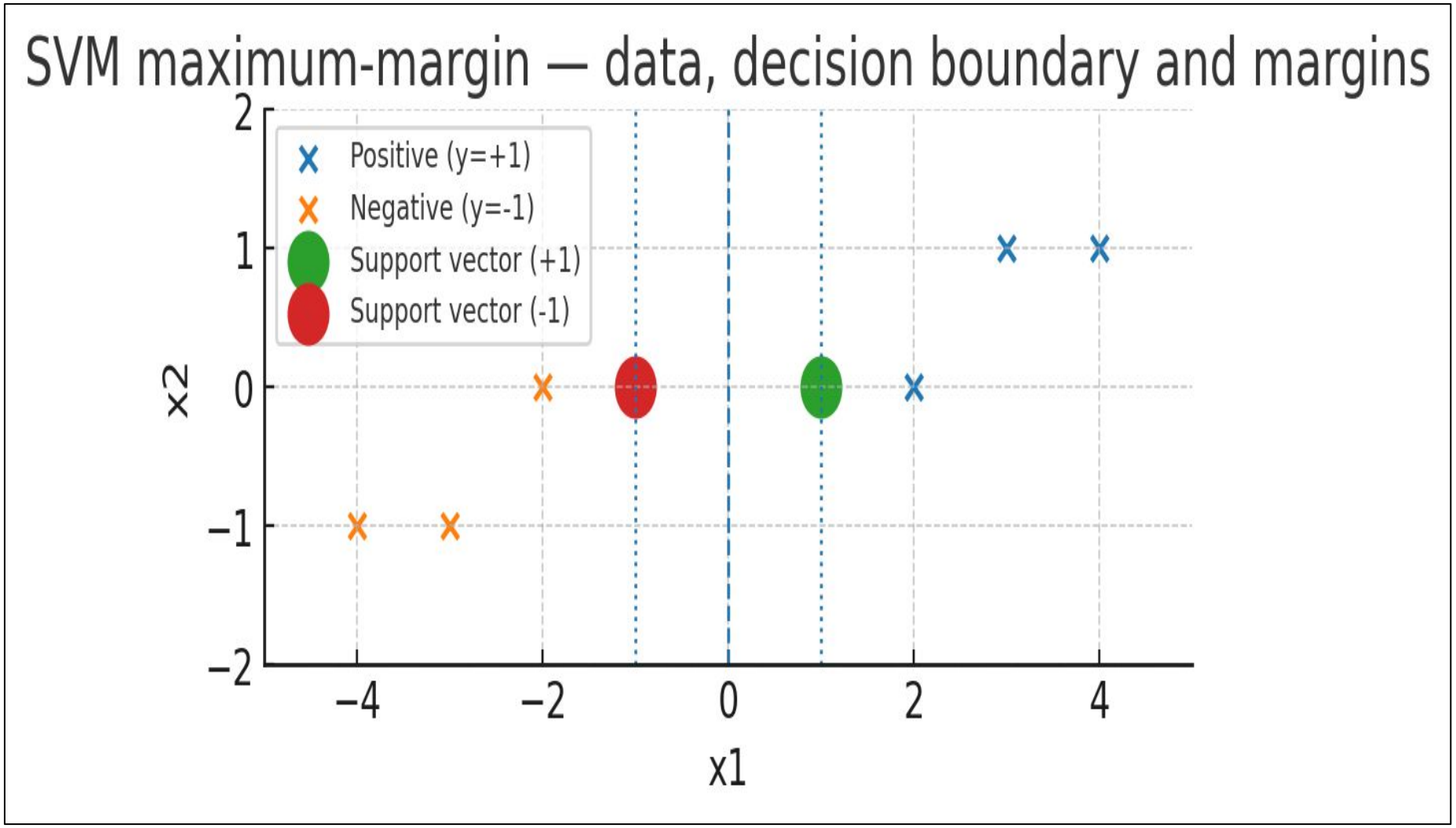
$$\text{distance (boundary}\to\text{margin)} = \frac{1}{\|w\|} = \frac{\sqrt{13}}{2} \approx 1.8028, \quad \text{margin width} = \frac{2}{\|w\|} = \sqrt{13} \approx 3.6056.$$

SVM maximum-margin example (symmetric dataset)

Dataset (labels chosen by sign of x1):

Positive class y=+1: (1,0), (2,0), (3,1), (4,1).

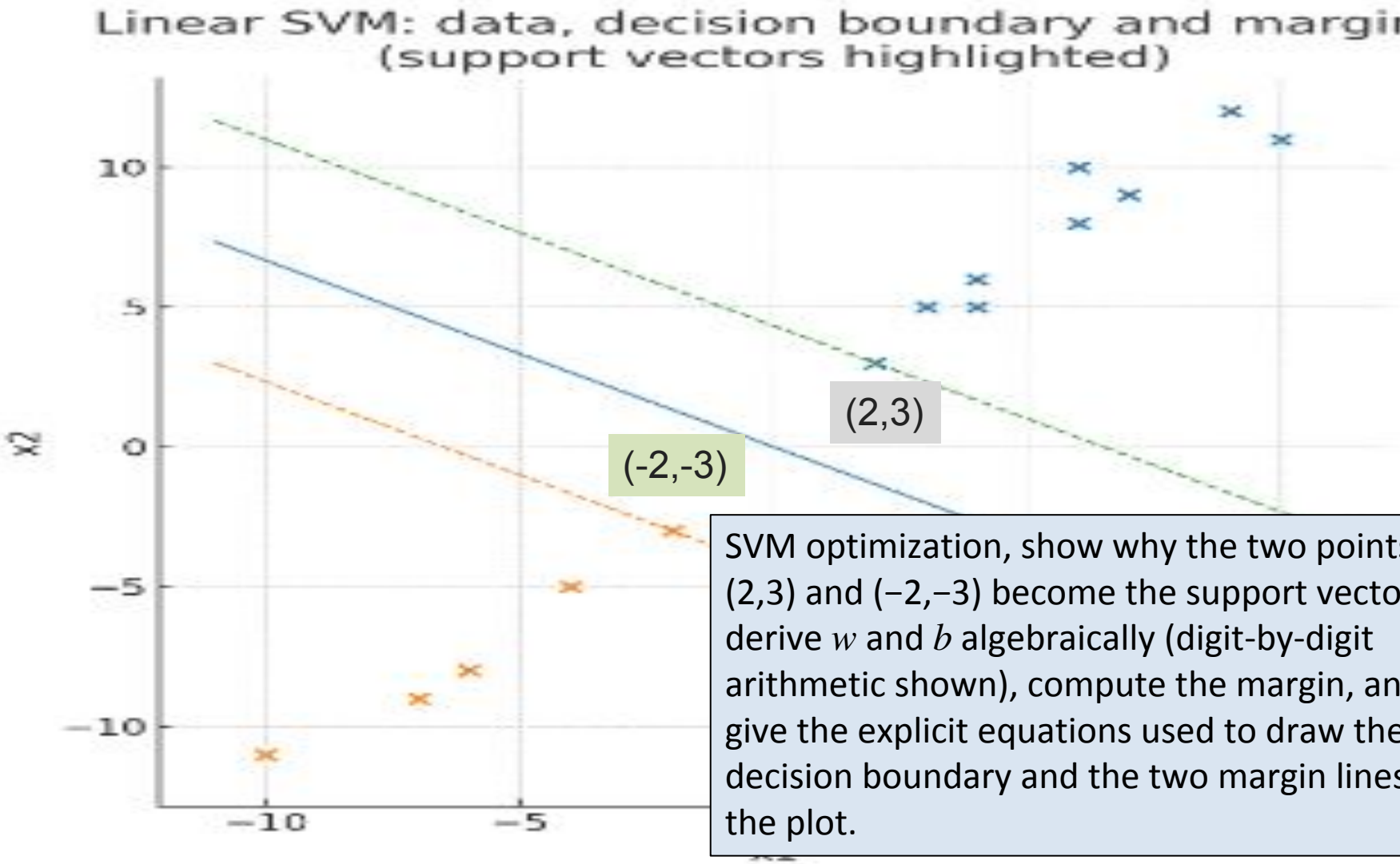
Negative class y=-1: (-1,0), (-2,0), (-3,-1), (-4,-1).



Prepare a complete worked out set of SVM - Maximum margin hyperplanes with symmetric data points (1,1), (2,2), (3,3), (4,4) and (-1,-1),(-2,-2),(-3,-3),(-4-4)

Prepare a complete worked out set of SVM - Maximum margin hyperplanes with symmetric data points (1,2),(2,3),(3,4),(4,5),(-1,-2),(-2,-3),(-3,-4),(-4,-5)

Prepare a complete worksheet of SVM - Maximum margin hyperplanes with asymmetric data points (2,3),(4,5),(6,8),(7,9) , (10,11), (3,5), (4,6), (6,10), (9,12), (-2,-3) (-4,-5),(-6,-8),(-7,-9),(-10,-11)



SVM optimization, show why the two points (2,3) and (−2,−3) become the support vectors, derive w and b algebraically (digit-by-digit arithmetic shown), compute the margin, and give the explicit equations used to draw the decision boundary and the two margin lines in the plot.

1 — Problem setup and labelling 14 points.

- Positive class $y = +1$:

  $(2,3), (4,5), (6,8), (7,9), (10,11), (3,5), (4,6), (6,10), (9,12)$ — 9 points.
- Negative class $y = -1$:

  $(-2,-3), (-4,-5), (-6,-8), (-7,-9), (-10,-11)$ — 5 points.

We assume the dataset is linearly separable (it is), so we use **hard-margin linear SVM**.

Characteristics of SVM

- The learning problem is formulated as a convex optimization problem
  - Efficient algorithms are available to find the global minima
  - Many of the other methods use greedy approaches and find locally optimal solutions
  - High computational complexity for building the model

- Robust to noise
- Overfitting is handled by maximizing the margin of the decision boundary,
- SVM can handle irrelevant and redundant better than many other techniques
- The user needs to provide the type of kernel function and cost function
- Difficult to handle missing values