UNIT-IV

(Support Vector Machine)

Topics :

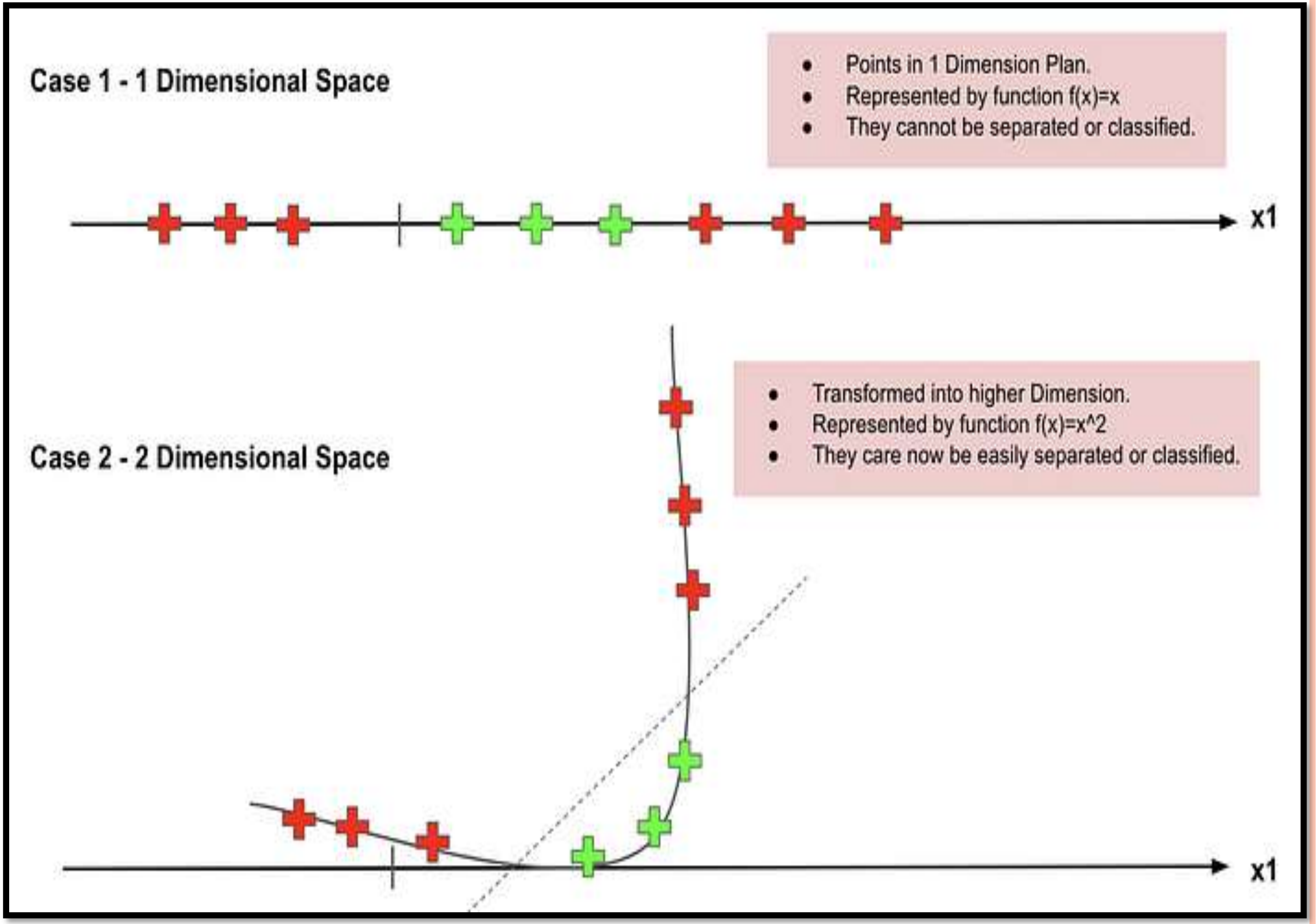
- **Support Vector Machine**
  - **Introduction to Support Vector Machines**
  - **Linear Support Vector Machines.**
  - **Non Linear Support Vector Machines**

Support Vector Machine

- **Support Vector Machine or SVM** is one of the most **popular Supervised Learning algorithms**, which is used for **Classification as well as Regression problems**.
- primarily, it is used for **Classification problems** in Machine Learning.
- The **objective of SVM algorithm** is to find a **hyperplane in an N-dimensional space** that **distinctly classifies the data points.**
- Here, The **dimension of the hyperplane** depends upon the **number of features.**

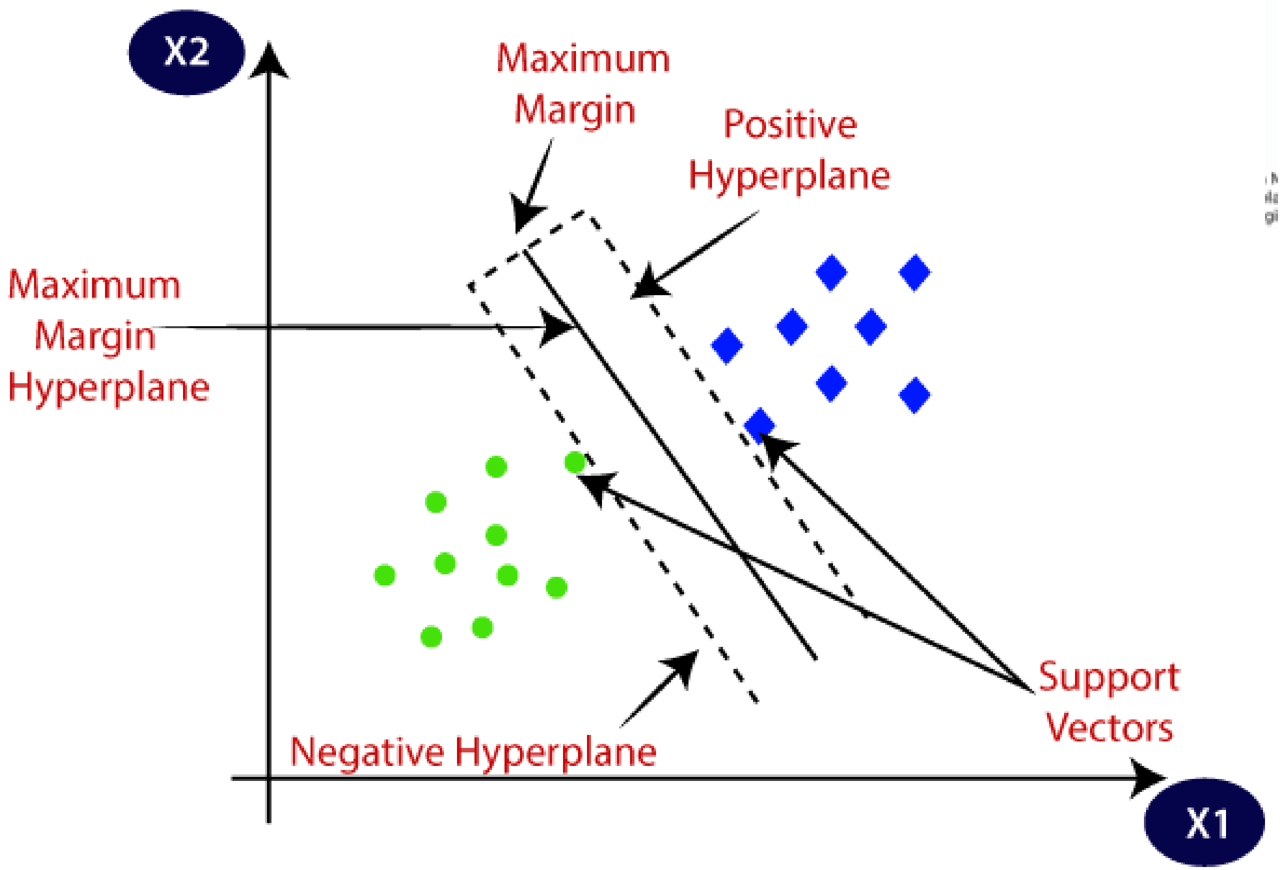
- The **goal of the SVM algorithm** is to create the **best line or decision boundary** that can segregate **n-dimensional space into classes**.
- so that we can easily put the **new data point in the correct category in the future.**
- This **best decision boundary** is called a **hyperplane.**
- If the **number of input features is two**, then the hyperplane is **just a line**.
- If the **number of input features is three**, then the hyperplane **becomes a 2-D plane.** It **becomes difficult** to imagine when the number of **features exceeds three.**

Contd..

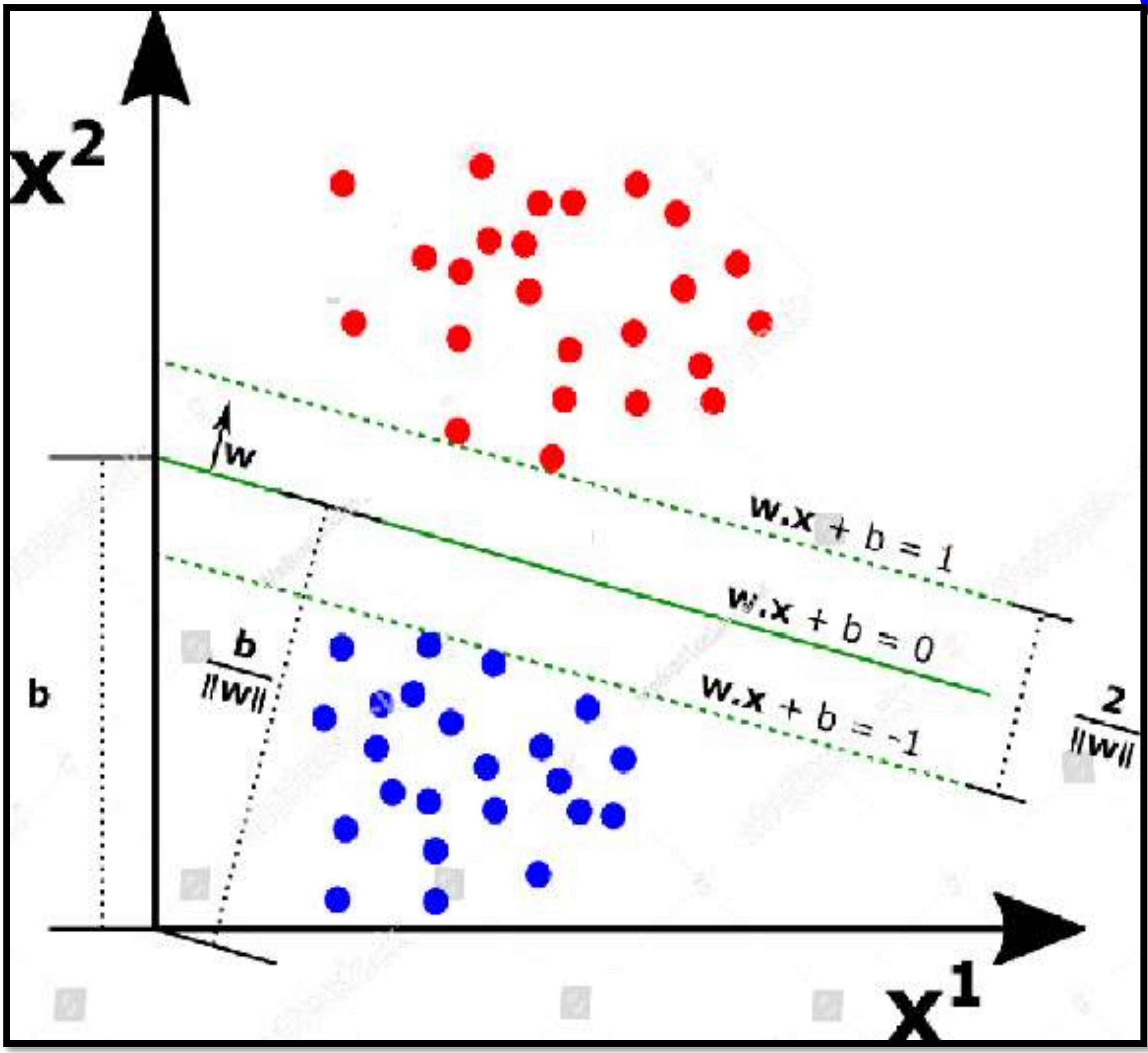

Case 1 - 1 Dimensional Space
Points in 1 Dimension Plan.
Represented by function f(x)=x
They cannot be separated or classified.
x1
Case 2 - 2 Dimensional Space
Transformed into higher Dimension.
Represented by function f(x)=x^2
They care now be easily separated or classified.
x1

- SVM chooses the **extreme points/vectors** that help in **creating the hyperplane**.
- These extreme cases are called as **support vectors**, and hence algorithm is termed as **Support Vector Machine.**
- Here, **Value of each feature** is also the value of the **specific coordinate**. Then, we find the **ideal hyperplane that differentiates between the two classes.**
- **Support vectors** are special because they are the **training points** that define the **maximum margin of the hyperplane** to the **data set** and they therefore determine the **shape of the hyperplane**

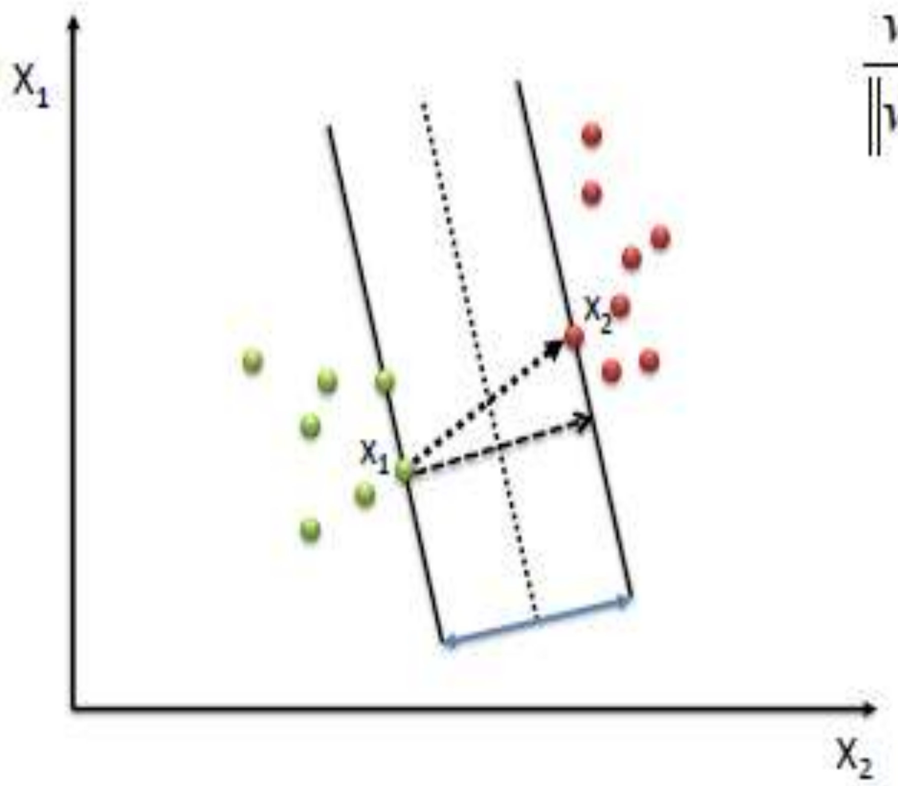
➢ Consider the **below diagram** in which there are **two different categories** that are classified using a **decision boundary or hyperplane:**



Contd..



- To define an **optimal hyperplane** we need to **maximize the width of the margin (*w*).**



$$\frac{w}{\|w\|}\cdot(x_2 - x_1) = \text{width} = \frac{2}{\|w\|}$$

$$w \cdot x_2 + b = 1$$

$$w \cdot x_1 + b = -1$$

$$w \cdot x_2 + b - w \cdot x_1 - b = 1 - (-1)$$

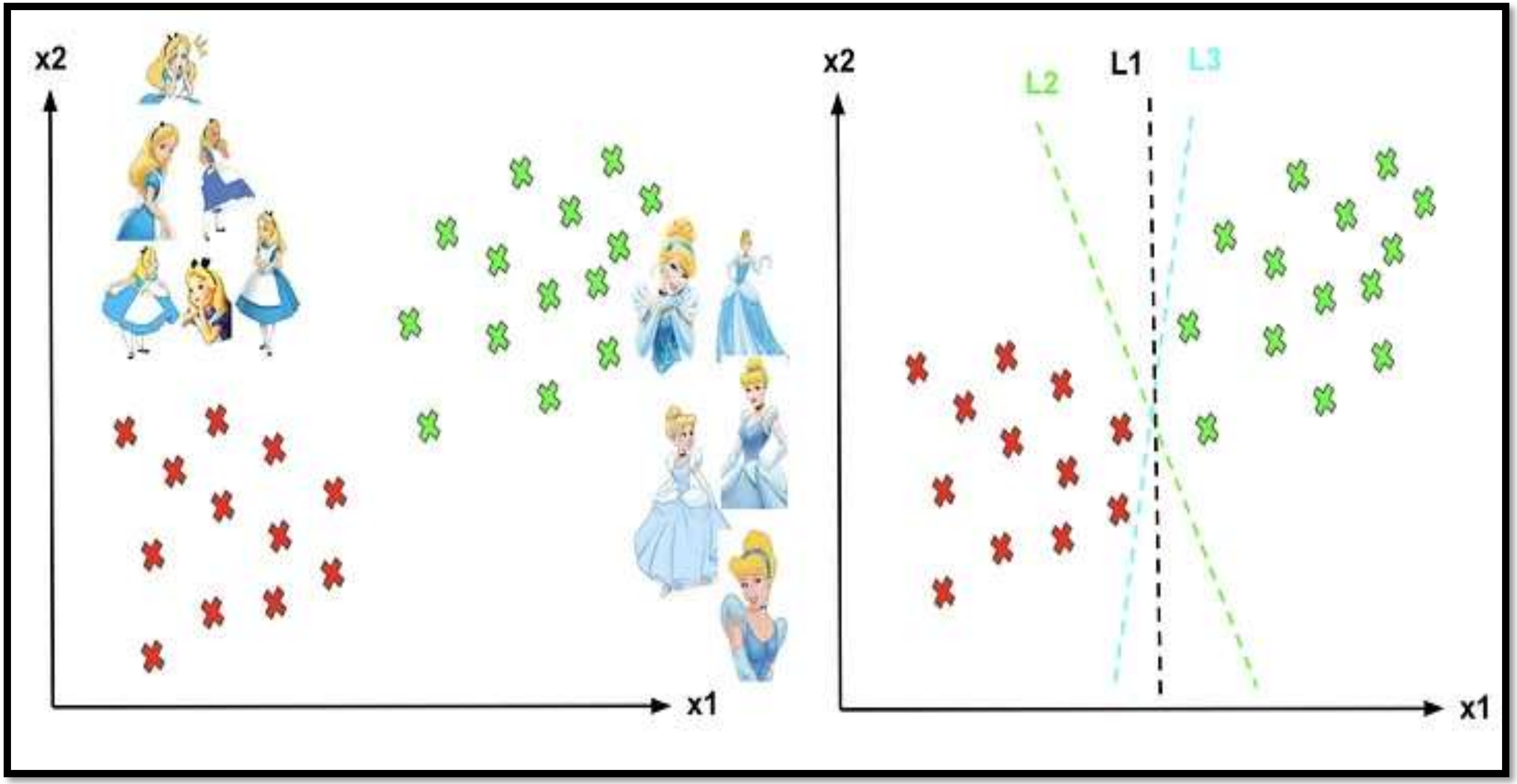
$$w \cdot x_2 - w \cdot x_1 = 2$$

$$\frac{w}{\|w\|}(x_2 - x_1) = \frac{2}{\|w\|}$$

EXAMPLE

- Imagine the **labelled training set** are **two classes of data points (two dimensions): Alice and Cinderella**.
- To **separate the two classes**, there are so many **possible options of hyperplanes that separate correctly.**
- As shown in the **graph below**, we can achieve exactly the same result using **different hyperplanes (L1, L2, L3).**

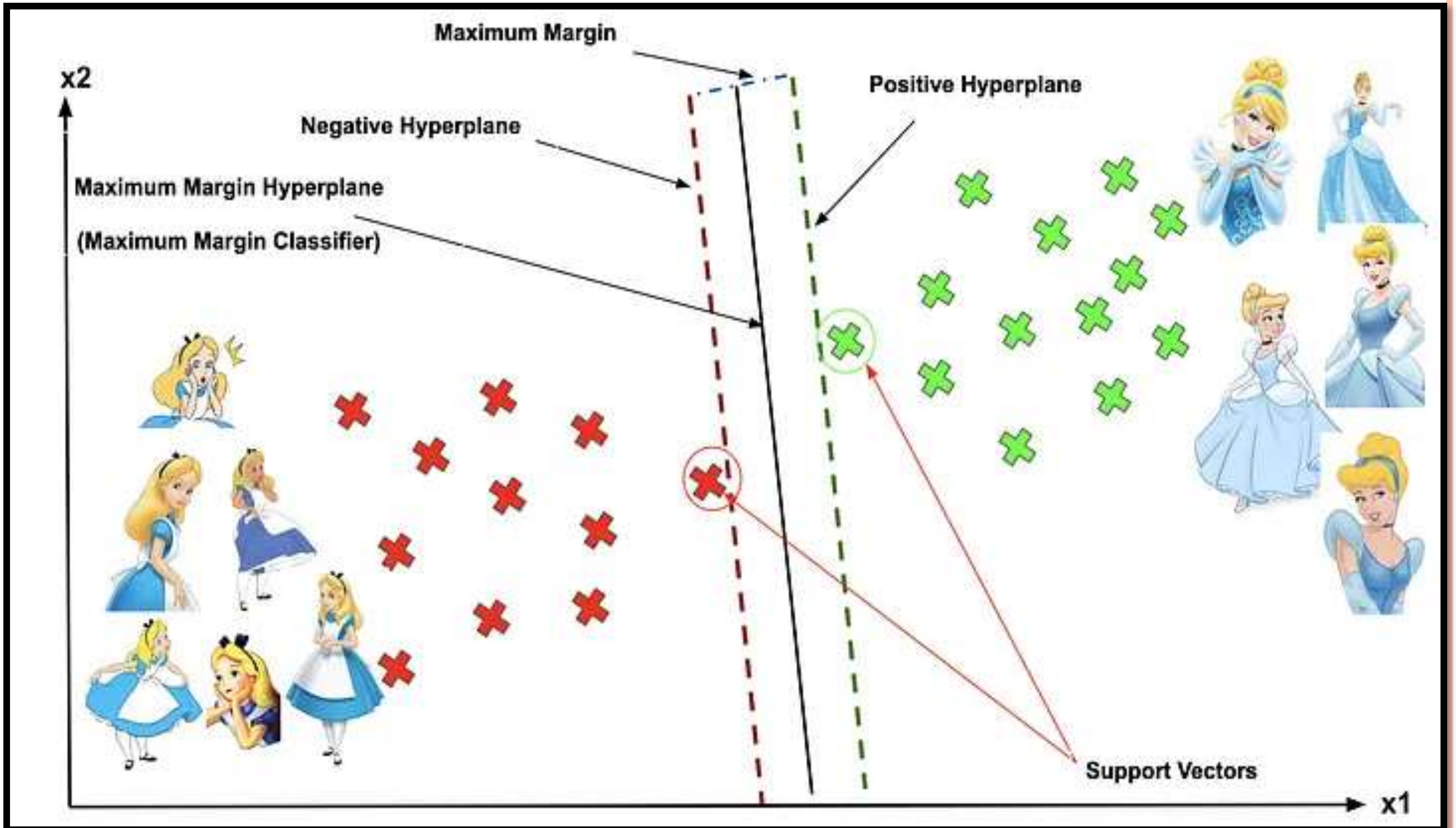
Contd..



Different hyperplanes (L1, L2, L3).

Contd..

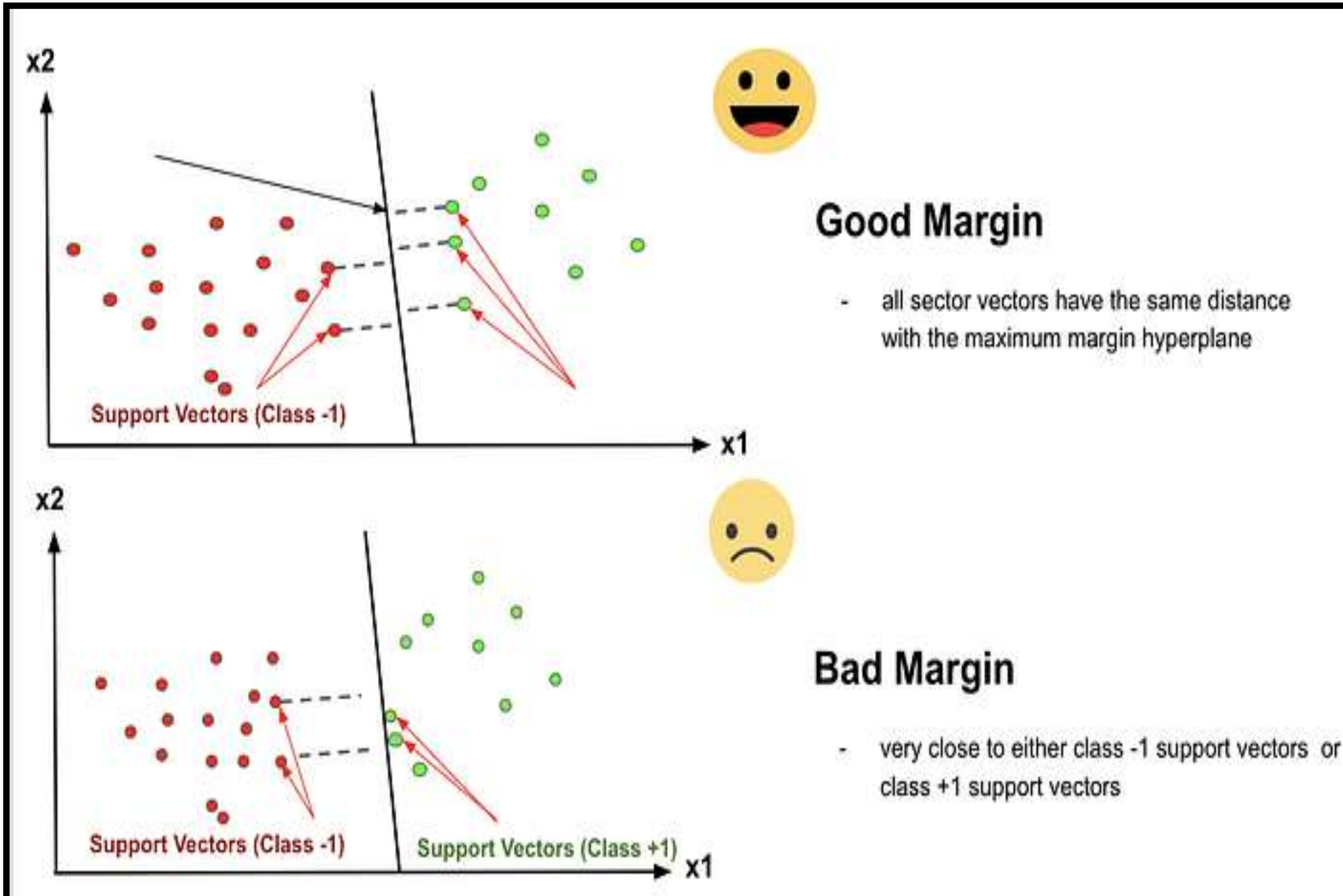
Q: **How can we decide a separating line for the classes? Which hyperplane shall we use?**



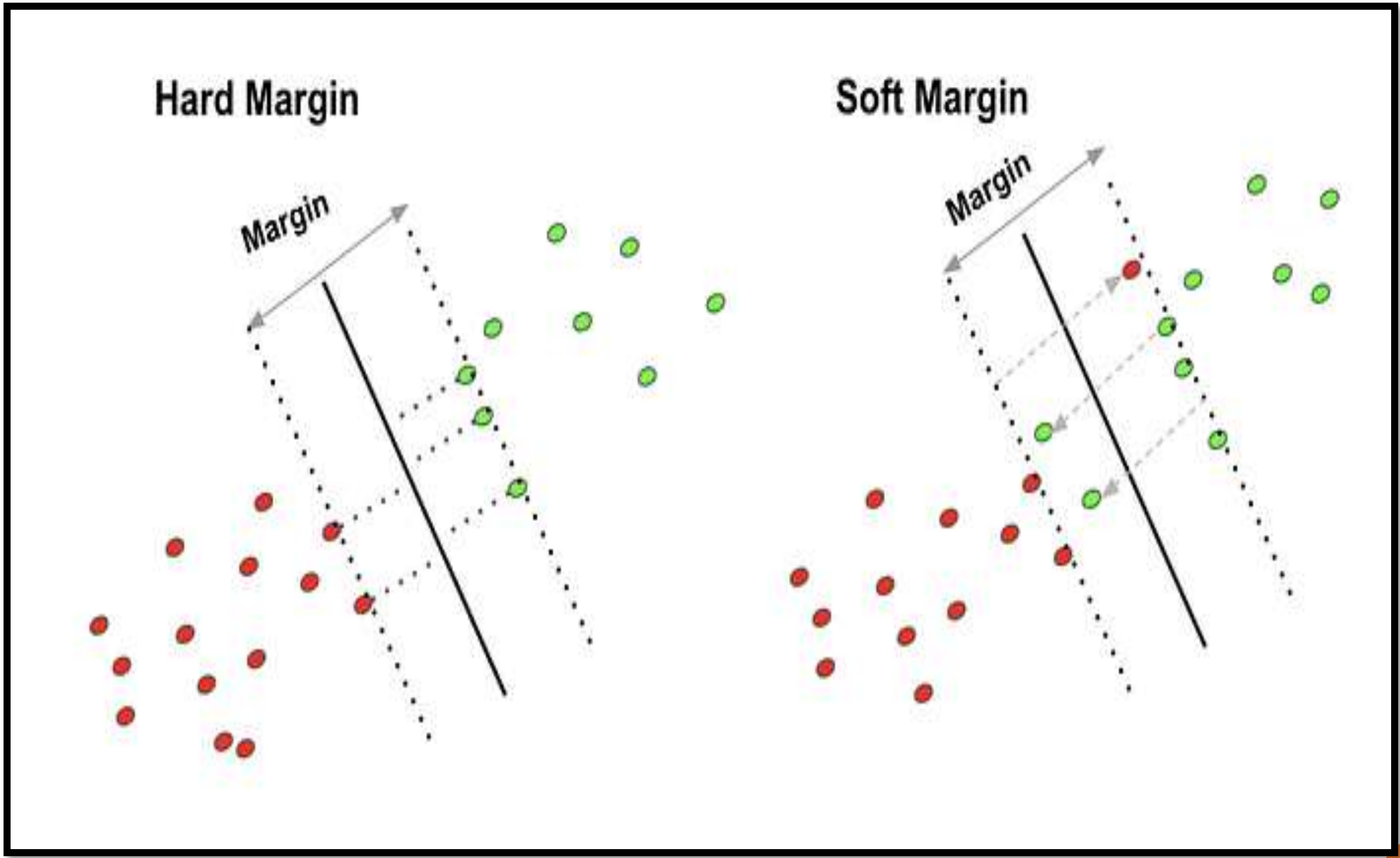
- The **vector points** **closest to the hyperplane** are known as the **support vector points** because only these **two points are contributing to the result of the algorithm**, and **other points are not.**
- If a **data point is not a support vector,** removing it has no effect on the model. On the other hand, deleting the **support vectors will then change the position of the hyperplane**.

- The **distance of the vectors from the hyperplane** is called the **margin,** which is a **separation of a line to the closest class points.**
- We would like to choose **a hyperplane that maximizes the margin between classes.** The graph below shows what good margin and bad margin are.

Contd..


x2
x1
Support Vectors (Class -1)
Good Margin
- all sector vectors have the same distance with the maximum margin hyperplane
x2
x1
Support Vectors (Class -1)
Support Vectors (Class +1)
Bad Margin
- very close to either class -1 support vectors or class +1 support vectors

Contd..



- We try to **fit a linear line** that can **separate the two classes** and **make the distances** from **either class is maximized.**
- **The line is represented with**

  $w*x + b = 0$

  For label 1, $w*x + b \geq 1$

  For label -1, $w*x + b \leq -1$
- The **distances of the gap in-between two classes** is **$2/|w|$**

we want to maximize this distance.

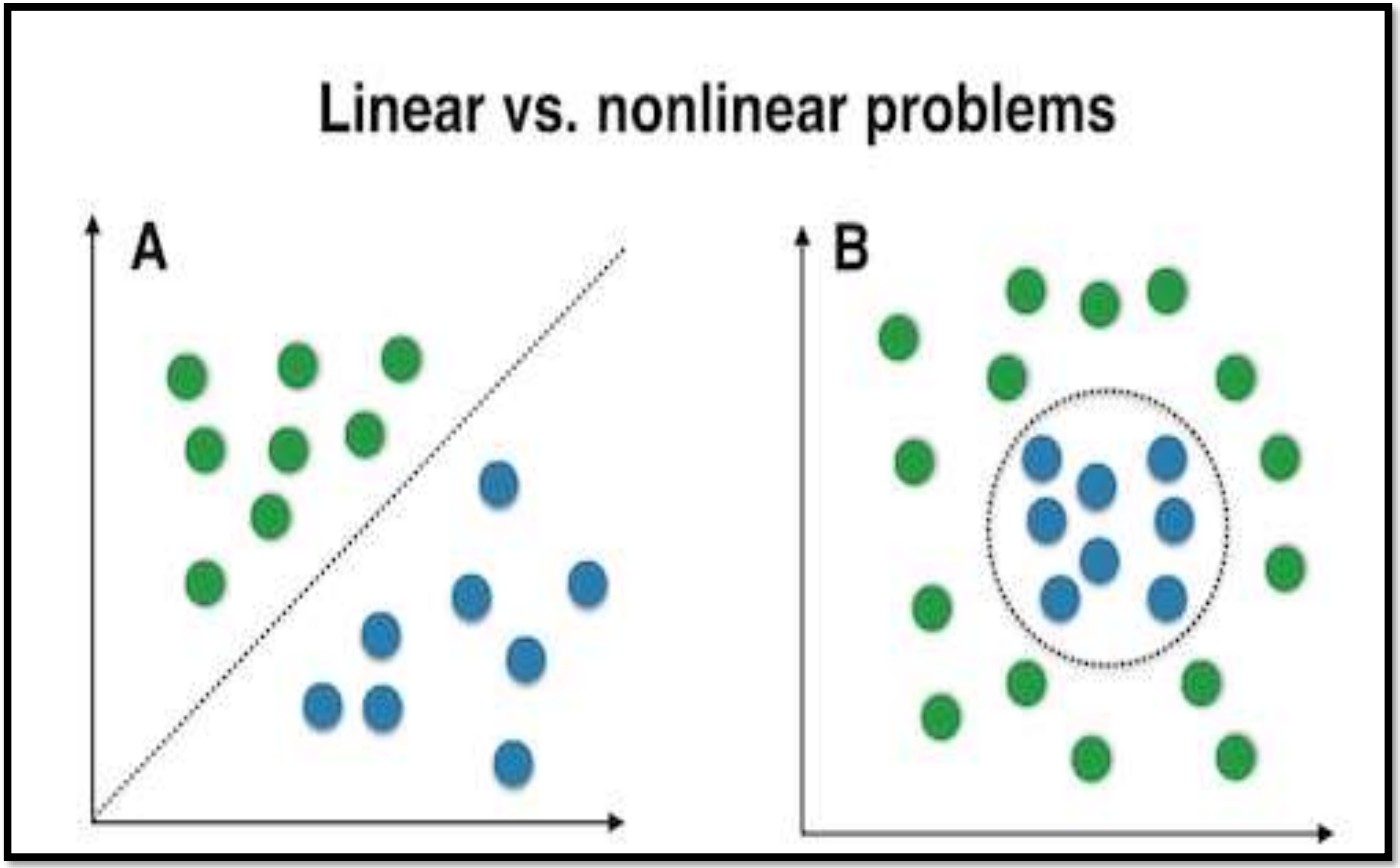
- If we want to find a line that **perfectly separates the two classes,** we call this type of **SVM cost function** as **Hard-Margin cost function.**

Types of SVM

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for **linearly separable data,** which means **if a dataset can be classified into two classes by using a single straight line**, then such data is termed as **linearly separable data**, and **classifier is used called as Linear SVM classifier.**
- **Non-linear SVM:** Non-Linear SVM is used **for non-linearly separated data,** which means if a **dataset cannot be classified by using a straight line,** then such data is termed as non-linear data and classifier used is called as **Non-linear SVM classifier.**

CONTD..



ADVANTAGES

- SVM works very well with **higher-dimensional datasets.**
- SVM is one of the most **memory-efficient classification algorithms.**
- The clearer the **margin of separation between the categories**, the better the SVM works.
- Effective on datasets with multiple features, like financial or medical data.

DISADVANTAGES

- SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.

APPLICATIONS

- Hand Written Detection
- Image Classification
- Face Detection
- Bioinformatics
- Cancer Detection
- Sentimental Analysis
- Spam Detection

EXAMPLE

In the **Given Dataset**, we **have 4 are positively labeled data sets and 4 are negatively labeled data sets.**

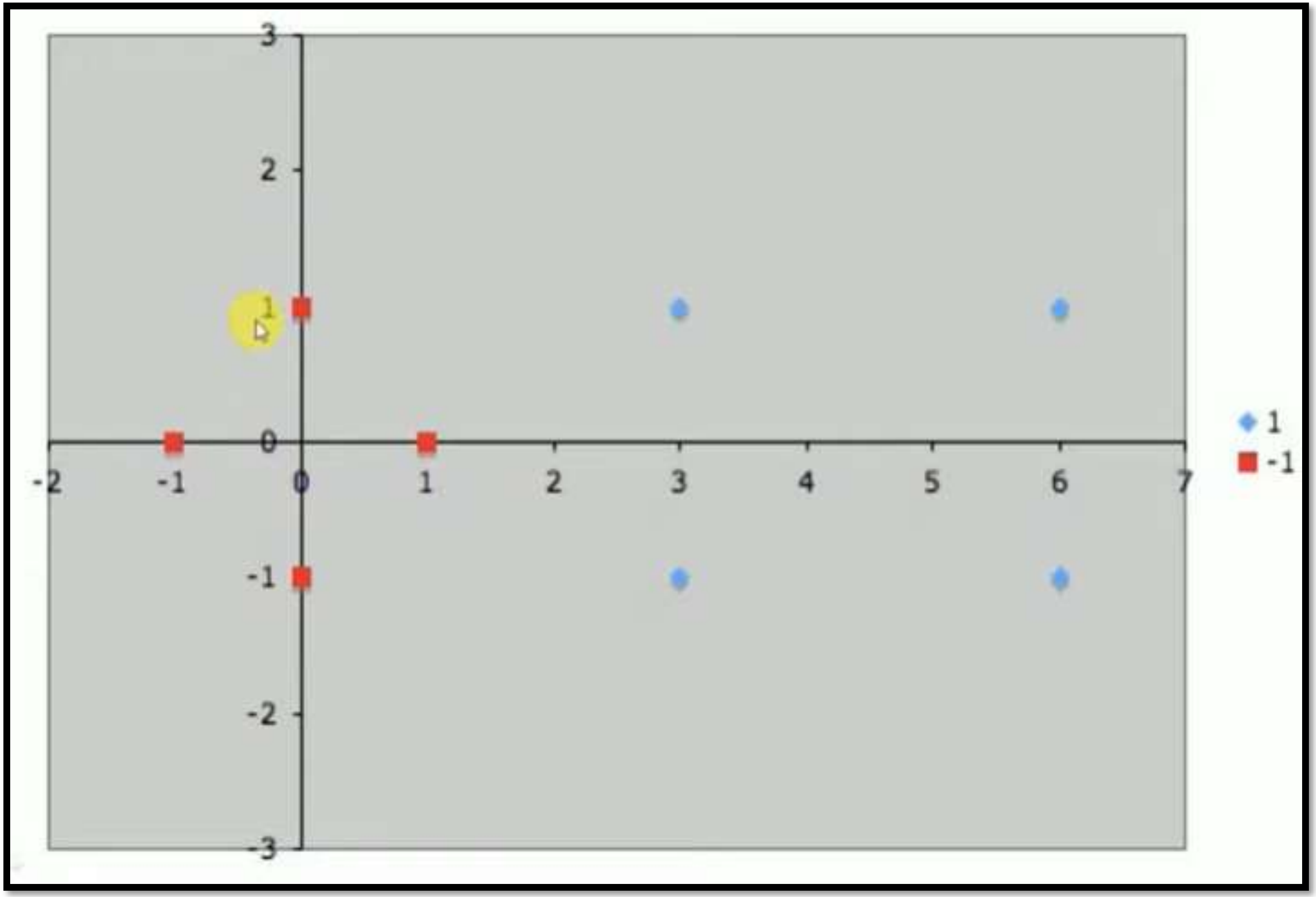
Suppose we are given the following positively labeled data points,

$$\left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix} \right\}$$

and the following negatively labeled data points,

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$$

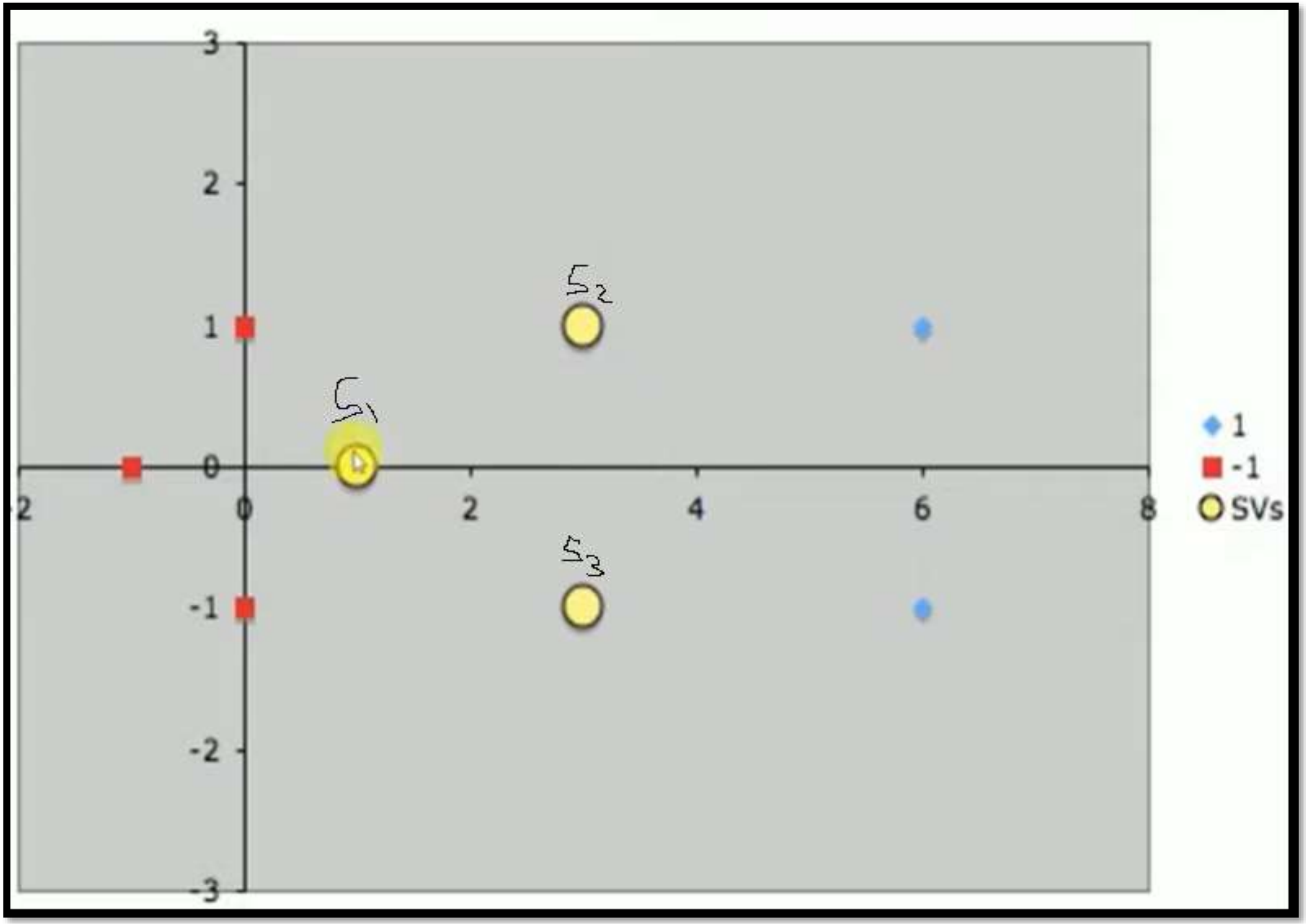
Next ,we need to plot the data points



Next, We need to identify the nearest data points on both the sides of these classes.

- By inspection, it should be obvious that there are **three** support vectors,

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \right\}$$



Here, S1 is Negatively labeled data set. And S2 and S3 are Positively Labeled Data sets.

CONTD..

Next we need to write the Hyper plane, we the bias=1 for wx+b Equation.

- Each vector is augmented with a 1 as a bias input
- So, $s_1 = \begin{pmatrix}1\\0\end{pmatrix}$, then $\tilde{s_1} = \begin{pmatrix}1\\0\\1\end{pmatrix}$
- Similarly,
- $s_2 = \begin{pmatrix}3\\1\end{pmatrix}$, then $\tilde{s_2} = \begin{pmatrix}3\\1\\1\end{pmatrix}$ *and* $s_3 = \begin{pmatrix}3\\-1\end{pmatrix}$, then $\tilde{s_3} = \begin{pmatrix}3\\-1\\1\end{pmatrix}$

We need to calculate the 3 Variables. ie, α1, α2, α3.

Which will be used to calculate the Weight Vector.

$$\begin{aligned}
\alpha_1 \tilde{s_1} \cdot \tilde{s_1} + \alpha_2 \tilde{s_2} \cdot \tilde{s_1} + \alpha_3 \tilde{s_3} \cdot \tilde{s_1} &= -1 \\
\alpha_1 \tilde{s_1} \cdot \tilde{s_2} + \alpha_2 \tilde{s_2} \cdot \tilde{s_2} + \alpha_3 \tilde{s_3} \cdot \tilde{s_2} &= +1 \\
\alpha_1 \tilde{s_1} \cdot \tilde{s_3} + \alpha_2 \tilde{s_2} \cdot \tilde{s_3} + \alpha_3 \tilde{s_3} \cdot \tilde{s_3} &= +1
\end{aligned}$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} = 1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} = 1$$

We can simplify this Equation.

Here we will Apply the DOT Product to simplify the Equation

$$\alpha_1(1+0+1)+\alpha_2(3+0+1)+\alpha_3(3+0+1)=-1$$

$$\alpha_1(3+0+1)+\alpha_2(9+1+1)+\alpha_3(9-1+1)=1$$

$$\alpha_1(3+0+1)+\alpha_2(9-1+1)+\alpha_3(9+1+1)=1$$

$$2\alpha_1+4\alpha_2+4\alpha_3=-1$$

$$4\alpha_1+11\alpha_2+9\alpha_3=1$$

$$4\alpha_1+9\alpha_2+11\alpha_3=1$$

$$2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1 \quad \text{— (1)}$$
$$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1 \quad \text{— (2)}$$
$$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1 \quad \text{— (3)}$$

(2) − (3)

$$\begin{aligned} \cancel{4\alpha_1} + 11\alpha_2 + 9\alpha_3 &= 1 \\ \cancel{4\alpha_1} + 9\alpha_2 + 11\alpha_3 &= 1 \\ \hline 2\alpha_2 - 2\alpha_3 &= 0. \\ \alpha_2 - \alpha_3 &= 0. \\ \alpha_2 &= \alpha_3. \end{aligned}$$

(2) − (1) × 2

$$\begin{aligned} \cancel{4\alpha_1} + 11\alpha_2 + 9\alpha_3 &= 1 \\ \cancel{4\alpha_1} + 8\alpha_2 + 8\alpha_3 &= -2 \\ \hline 3\alpha_2 + \alpha_3 &= 3. \\ 3\alpha_2 + \alpha_2 &= 3 \\ 4\alpha_2 &= 3. \\ \alpha_2 &= \frac{3}{4} = \alpha_3 \end{aligned}$$

$$2\alpha_1 + \cancel{4} \times \frac{3}{\cancel{4}} + \cancel{4} \times \frac{3}{\cancel{4}} = -1$$
$$2\alpha_1 + 3 + 3 = -1$$
$$2\alpha_1 + 6 = -1$$
$$2\alpha_1 = -7$$
$$\alpha_1 = -7/2.$$
$$\alpha_2 = 3/4$$
$$\alpha_3 = 3/4.$$

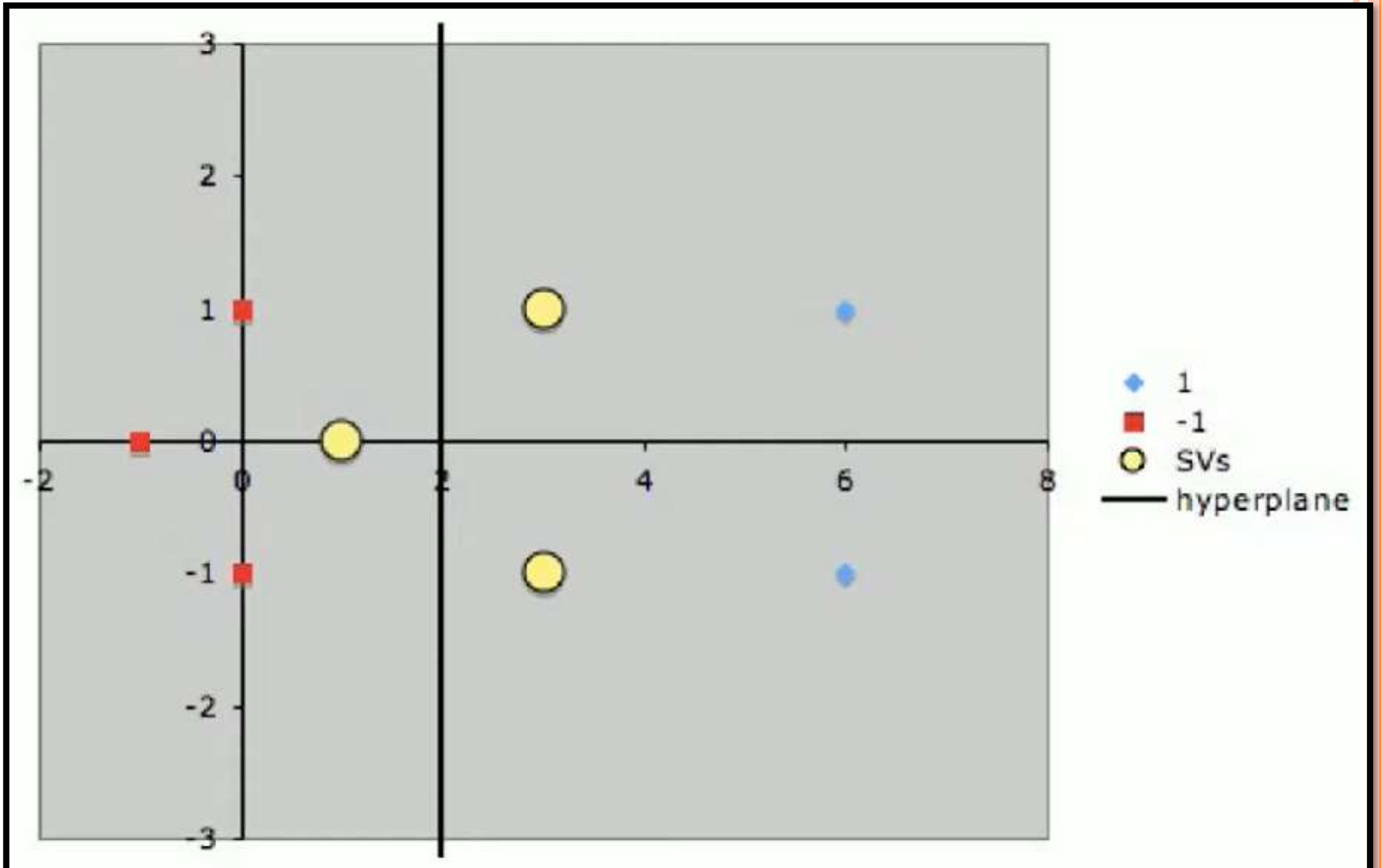
$$\alpha_1 = -3.5$$
$$\alpha_2 = 0.75$$
$$\alpha_3 = 0.75$$

Once we get the 3 variables, after that we need to calculate the Weight Vector.

$$\begin{aligned}
\tilde{w} &= \sum_i \alpha_i \tilde{s}_i \\
&= -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}
\end{aligned}$$

- Finally, remembering that our vectors are augmented with a bias.
- We can equate the last entry in $\tilde{w}$ as the hyperplane offset b and write the separating
- Hyperplane equation $\boldsymbol{y} = \boldsymbol{wx} + \boldsymbol{b}$
- with $\boldsymbol{w} = \binom{1}{0}$ and $\boldsymbol{b} = -2$.

Here we get, the line is (1,0) ie, the line is parallel to Y- Axis.
Suppose , If the Line is (0,1) ie, the line is parallel to X- Axis.
If the line is (1,1) , ie, the line is parallel to 45 Degrees.



THANK YOU

UNIT-IV
(SUPPORT VECTOR MACHINE)

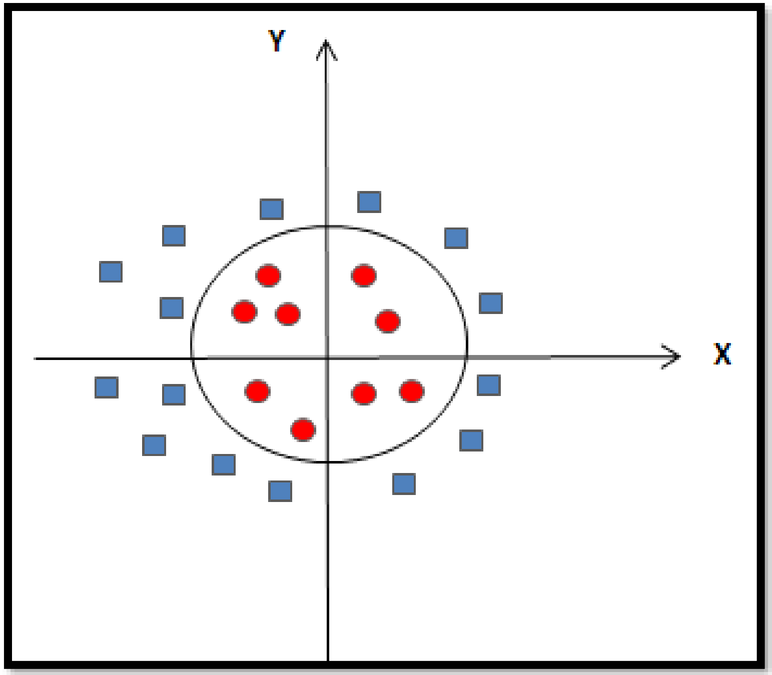
Topics :

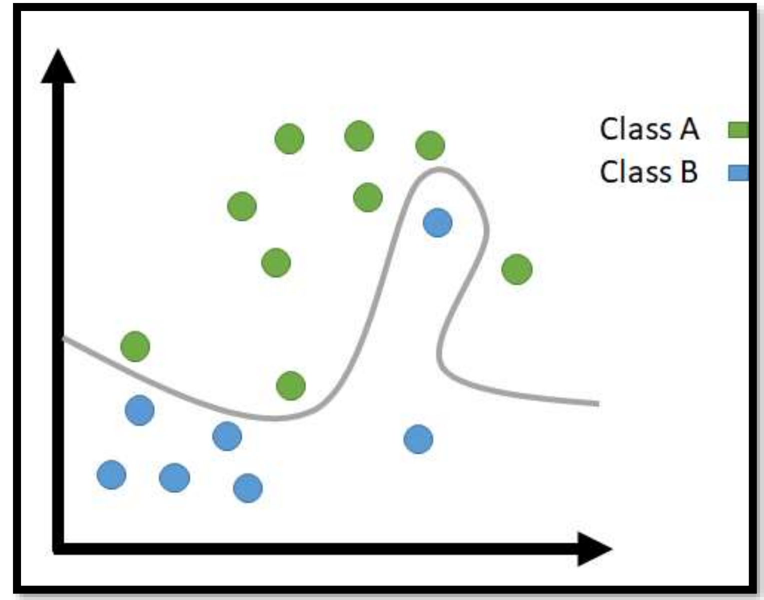
- **Support Vector Machine**
  - **Non Linear Support Vector Machines.**
  - **Example**

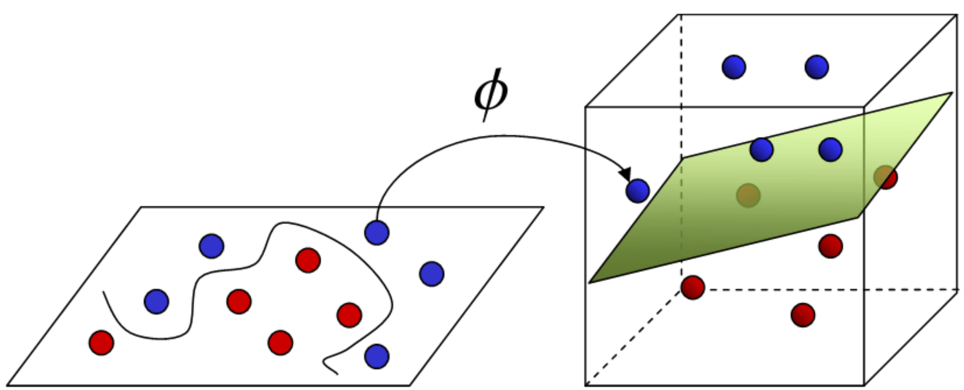
NON LINEAR SVM

- If **data is linearly arranged**, then we can **separate it by using a straight line**, but for **non-linear data,** we **cannot draw a single straight line.**
- **Imagine a case** - if there is no straight line (or hyper plane) which can separate two classes? In the **image shown below**, there is a **circle in 2D with red and blue data points** all over it such that **adjacent data points are of different colors.**

CONTD..

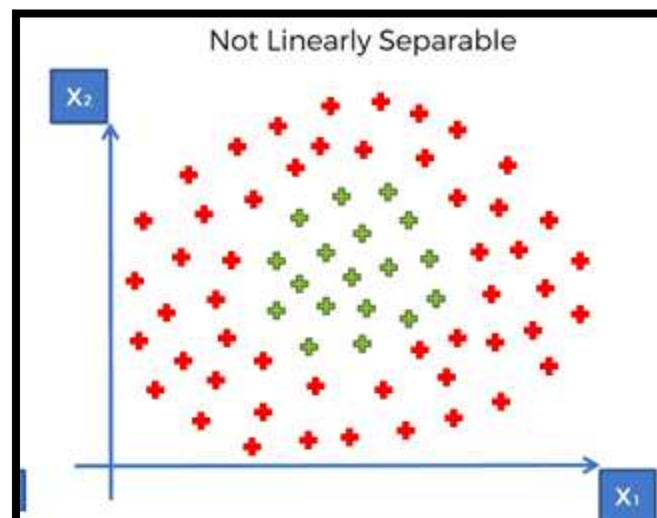






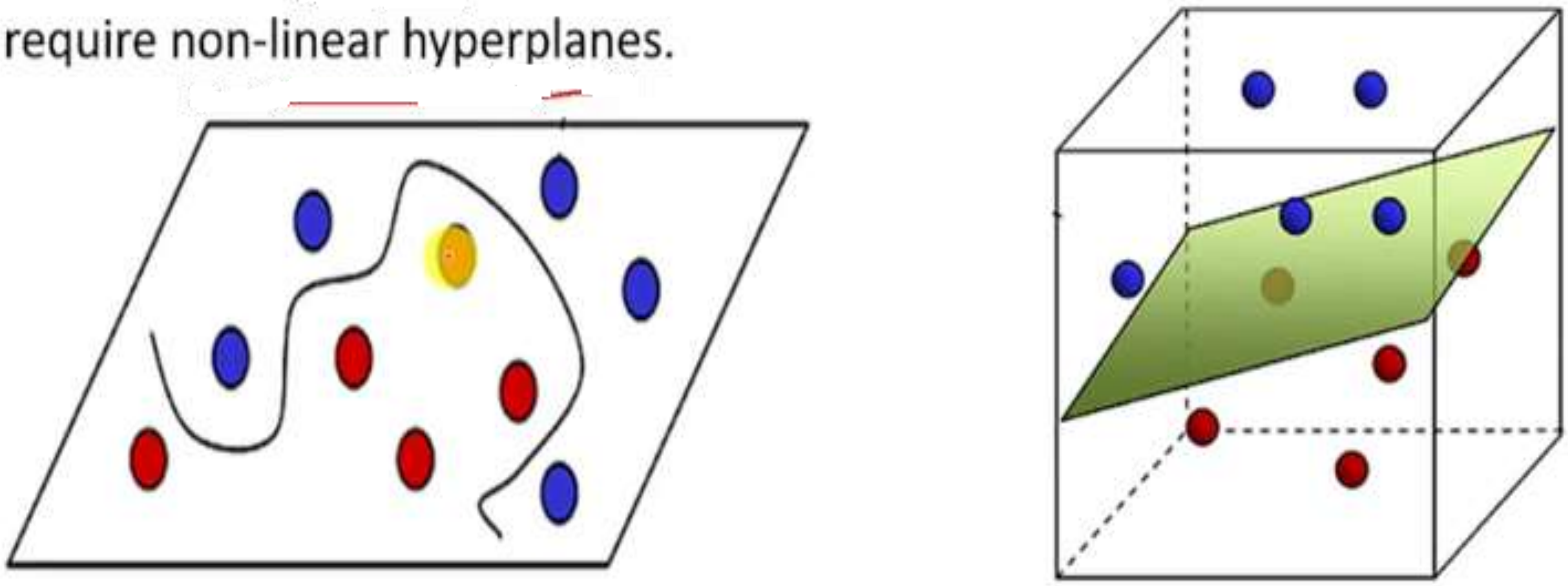
a) Input Space

b) Feature Space

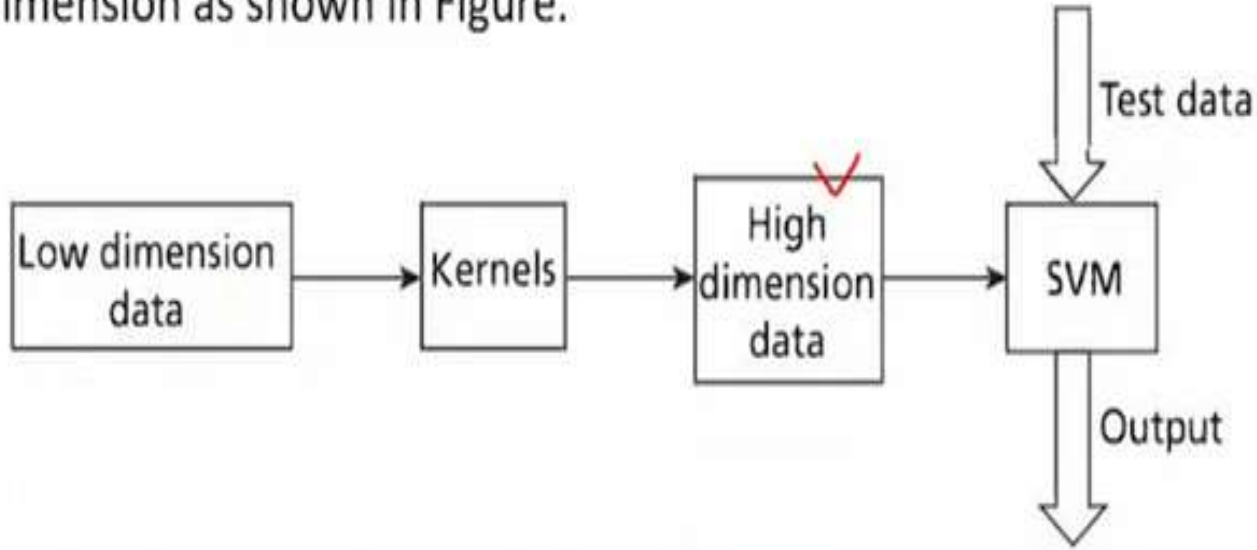


- So to **separate these data points**, we need **to add one more dimension.**
- For **linear data**, we have used **two dimensions x and y,** so for **non-linear data**, we will **add a third dimension z.**
- So ,here we use a **kernel function** to handle **non-linear separable data.**
- **Kernels** are used by **classification algorithms** to solve **non-linear classification problems.**
- The kernel function transforms the **data into a higher dimensional feature space** to make it possible to **perform the linear separation.**

- In machine learning applications, the data can be text, image, or video.
- So, there is a need to extract features from these data prior to classification.
- Hence, in the real world, many classification models are complex and mostly require non-linear hyperplanes.



- Kernels are a set of functions used to transform data from lower dimension to higher dimension and to manipulate data using dot product at higher dimensions.
- The use of kernels is to apply transformation to data and perform classification at the higher dimension as shown in Figure.



Contd..

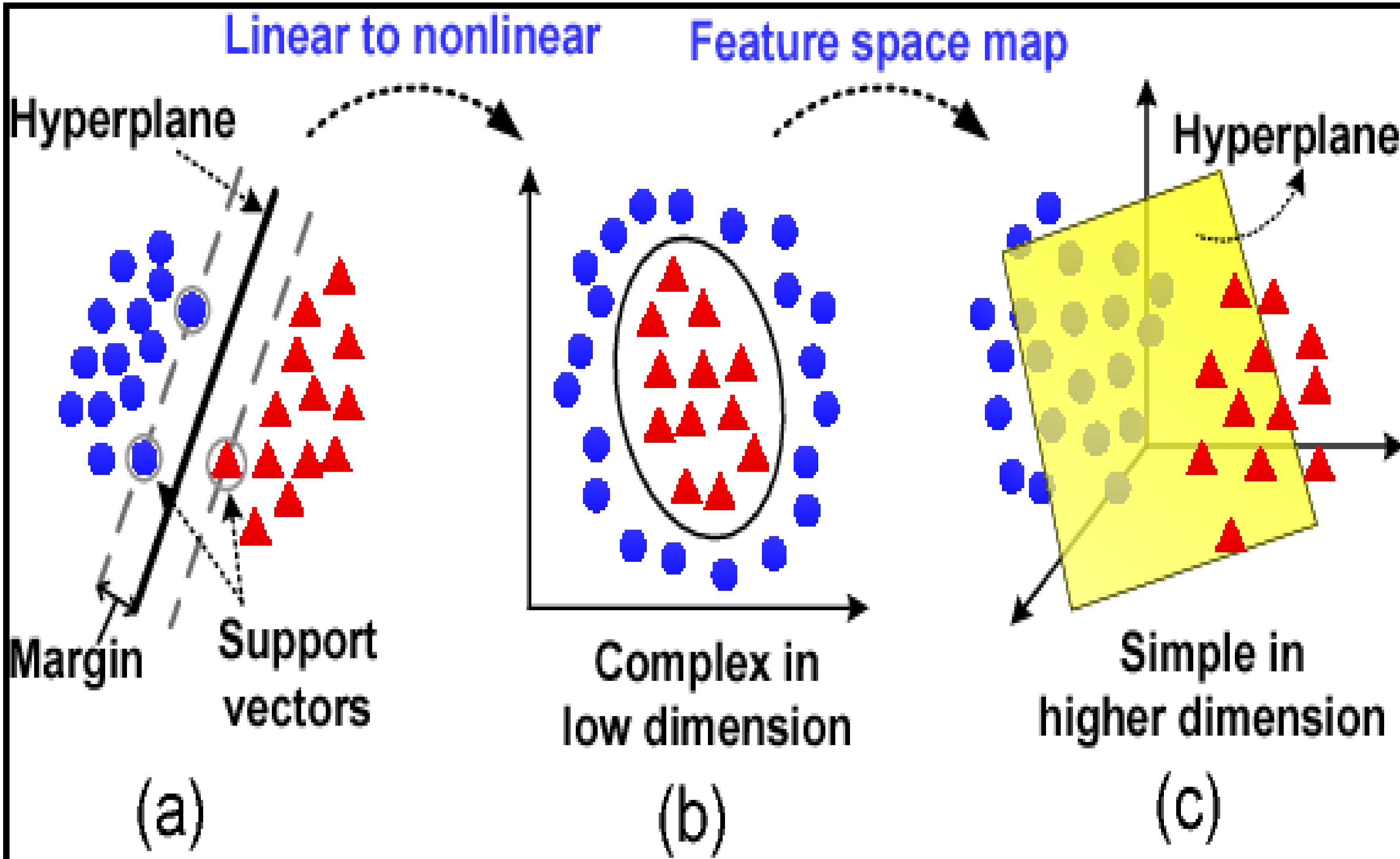
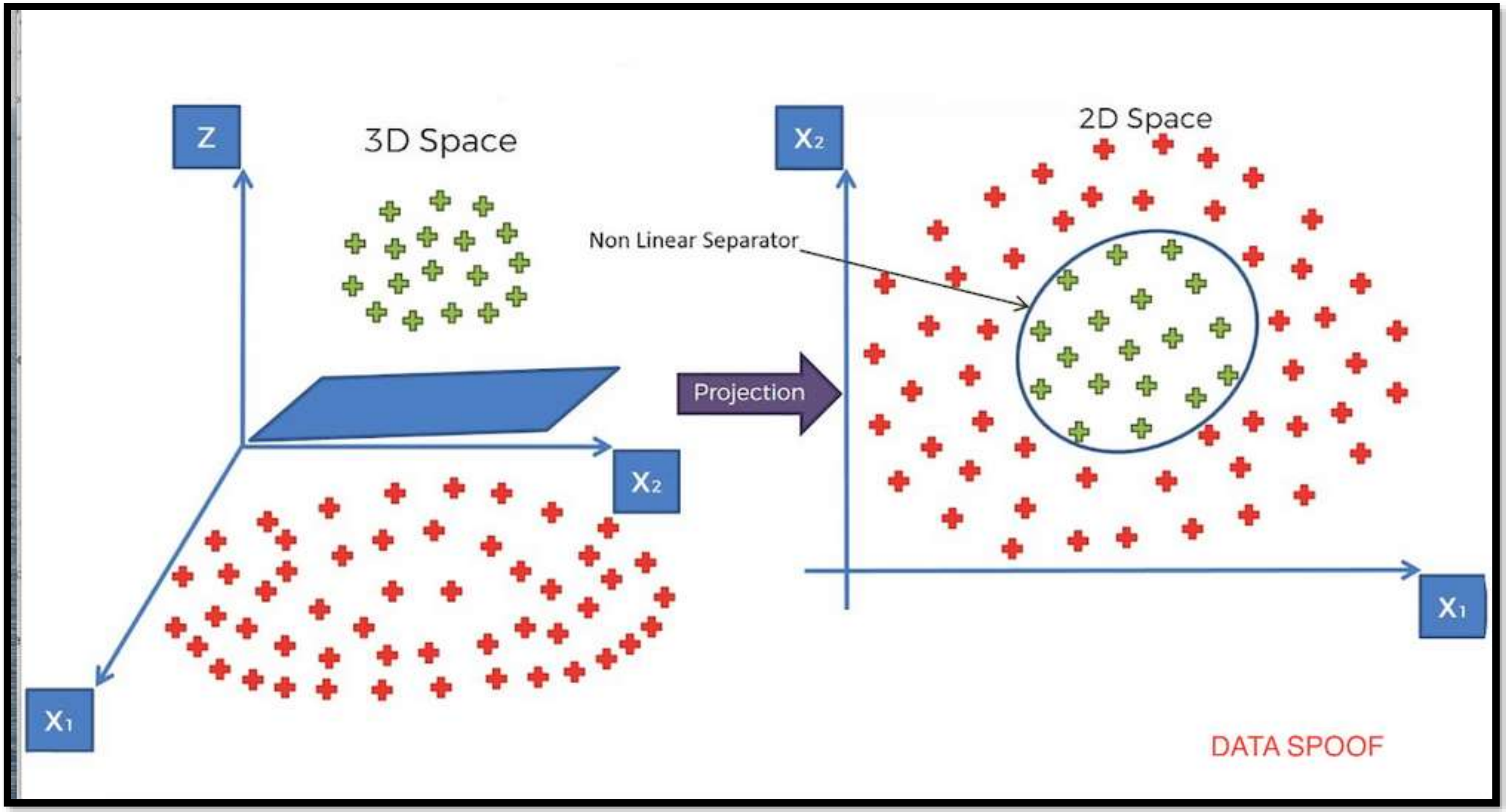


Fig. 1: SVM schematic diagram

Example



Z
3D Space
X₂
X₁
Projection
Non Linear Separator
2D Space
X₂
X₁
DATA SPOOF

CONTD..

- he kernel function is a function that may be expressed as the **dot product of the mapping function** (kernel method) and looks like this, $\mathbf{K(x_i,x_j) = Ø(x_i) \cdot Ø(x_j)}$
- There are **different types of Kernels** are available to convert non linear to linear.

  1. **linear**: u'*v
  2. **polynomial**: (gamma*u'*v + coef0)^degree
  3. **radial basis** (RBF) : exp(-gamma*|u-v|^2)
  4. **sigmoid** : tanh(gamma*u'*v + coef0)

- **RBF is generally the most popular one.**

Contd..

The mathematical formula behind RBF is:-

$$K(x, x') = e^{-\gamma \|x - x'\|^2}$$

Gamma is a scalar that defines how much influence a single training example (point) has.

EXAMPLE

In the **Given Dataset**, we **have 4 are positively labeled data sets and 4 are negatively labeled data sets.**

Suppose we are given the following positively labeled data points,

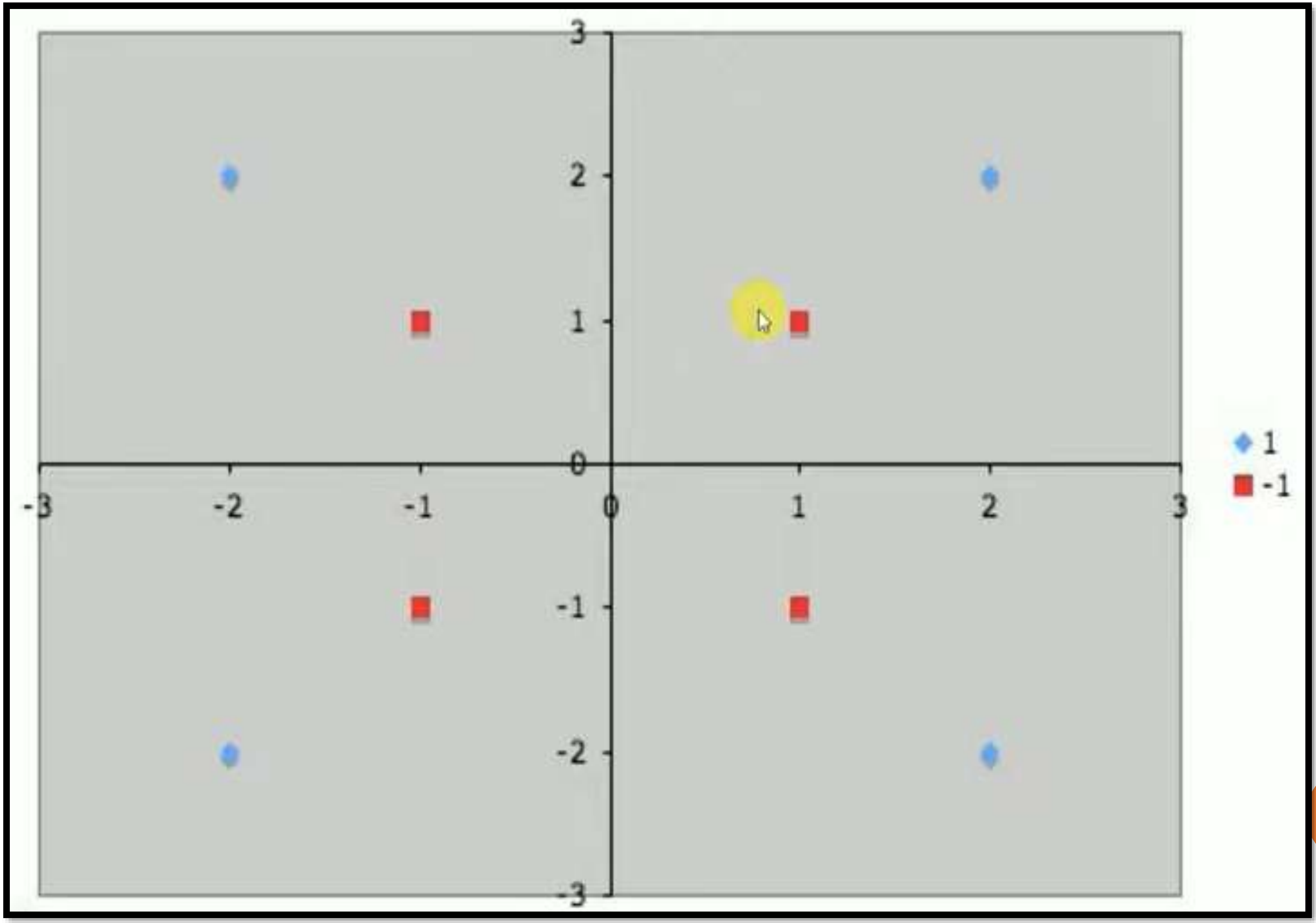
$$\left\{\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix}\right\}$$

and the following negatively labeled data points,

$$\left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}\right\}$$

CONTD..

Next ,we need to plot the data points



Here our Goal is to separate Hyper plane that accurately separately two classes.

- Therefore, we must use a nonlinear SVM (that is, we need to convert data from one feature space to another.

$$\Phi_1\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} 4 - x_2 + |x_1 - x_2| \\ 4 - x_1 + |x_1 - x_2| \end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} > 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{otherwise} \end{cases}$$

Here our Goal is to separate Hyper plane that accurately separately two classes by using the Mapping Function RBF.

$$\Phi_1\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} 4 - x_2 + |x_1 - x_2| \\ 4 - x_1 + |x_1 - x_2| \end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} > 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{otherwise} \end{cases}$$

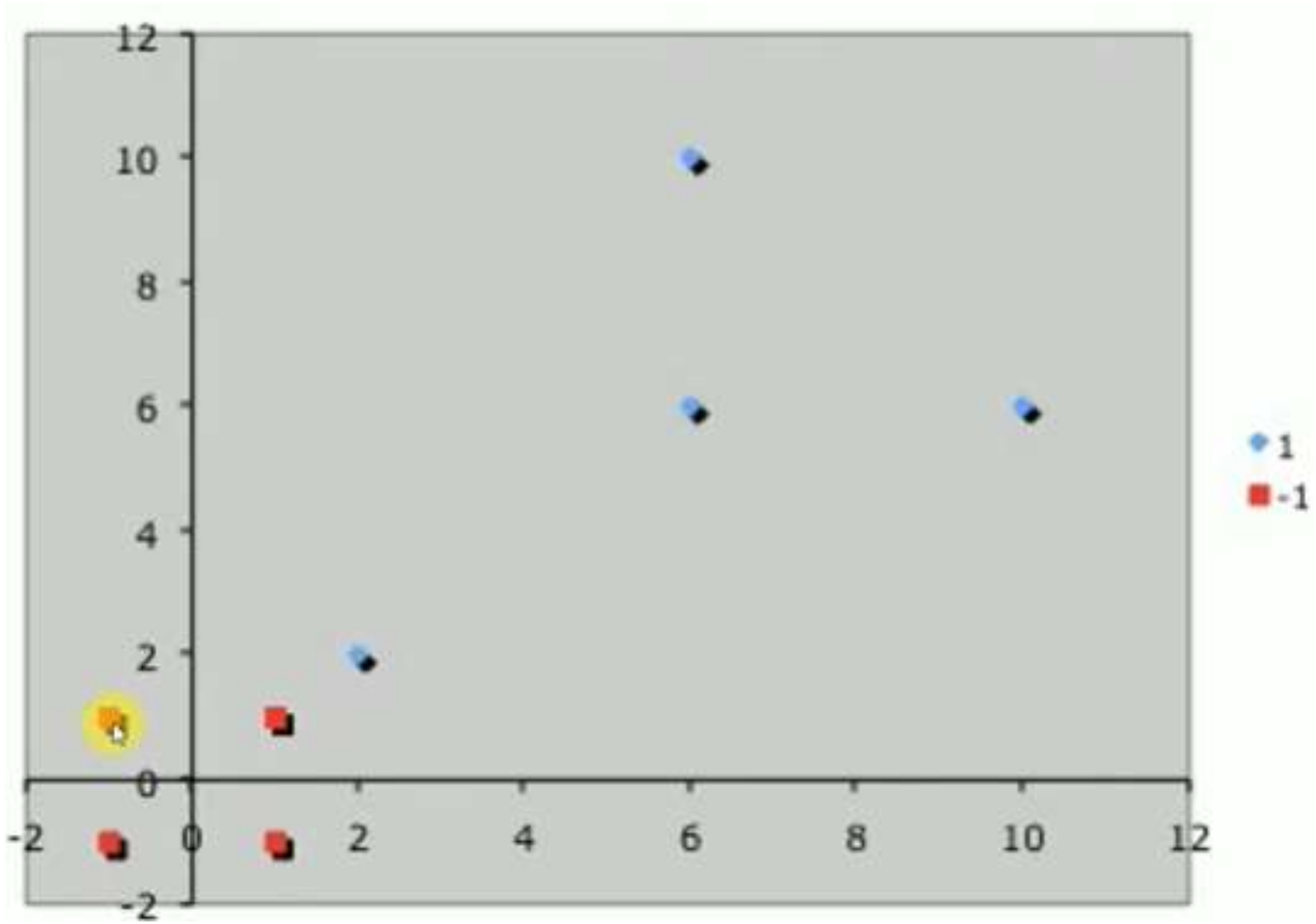
Positive Examples

$$\left\{\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix}\right\} \rightarrow \left\{\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 10 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 10 \end{pmatrix}\right\}$$

Negative Examples

$$\left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}\right\} \rightarrow \left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}\right\}$$

Next we can draw the Hyper plane by using updated Data Point



- Now we can easily identify the support vectors,

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, s_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\}$$

- Each vector is augmented with a 1 as a bias input

$$\tilde{s_1} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad \tilde{s_2} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

We need to calculate the 3 Variables. ie, α1, α2.

Which will be used to calculate the Weight Vector.

$$\alpha_1 \tilde{s_1} \cdot \tilde{s_1} + \alpha_2 \tilde{s_2} \cdot \tilde{s_1} = -1$$

$$\alpha_1 \tilde{s_1} \cdot \tilde{s_2} + \alpha_2 \tilde{s_2} \cdot \tilde{s_2} = +1$$

$$\alpha_1 \begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}2\\2\\1\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}2\\2\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}2\\2\\1\end{pmatrix}\begin{pmatrix}2\\2\\1\end{pmatrix} = 1$$

$$\alpha_1(1+1+1) + \alpha_2(2+2+1) = -1$$

$$\alpha_1(2+2+1) + \alpha_2(4+4+1) = 1$$

$$3\alpha_1 + 5\alpha_2 = -1$$

$$5\alpha_1 + 9\alpha_2 = 1$$

$$\alpha_1 = -7$$

$$\alpha_2 = 4$$

Once we get the 2 variables, after that we need to calculate the Weight Vector.

$$\tilde{w} = \sum_i \alpha_i \tilde{s}_i = -7\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 4\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

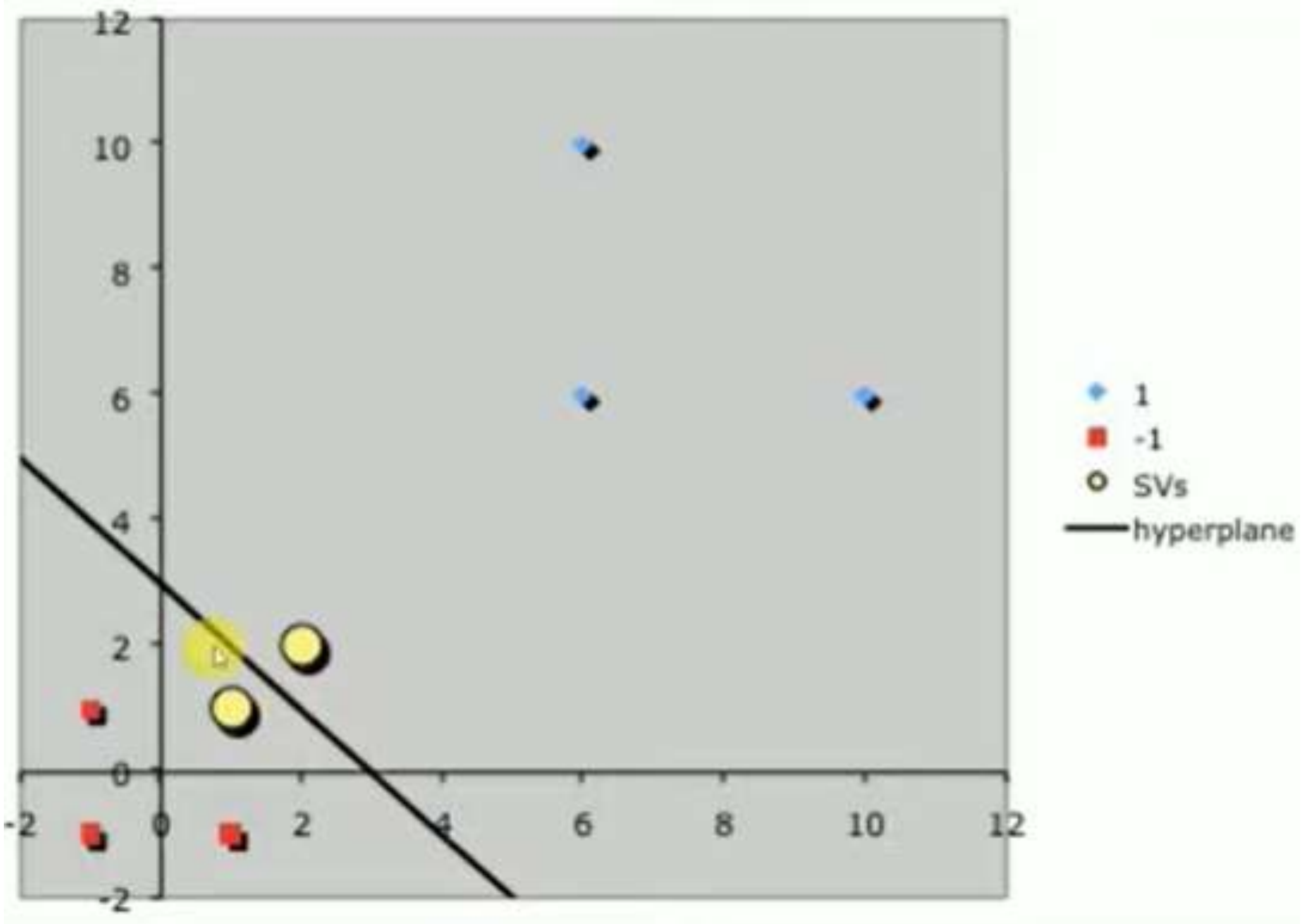
$$= \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}$$

- Finally, remembering that our vectors are augmented with a bias.
- We can equate the last entry in $\tilde{w}$ as the hyperplane offset b and write the separating
- Hyperplane equation $\boldsymbol{y} = \boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}$
- with $\boldsymbol{w} = \binom{1}{1}$ and $\boldsymbol{b} = -3$.

Here we get, the line is (1,0) ie, the line is parallel to Y- Axis.

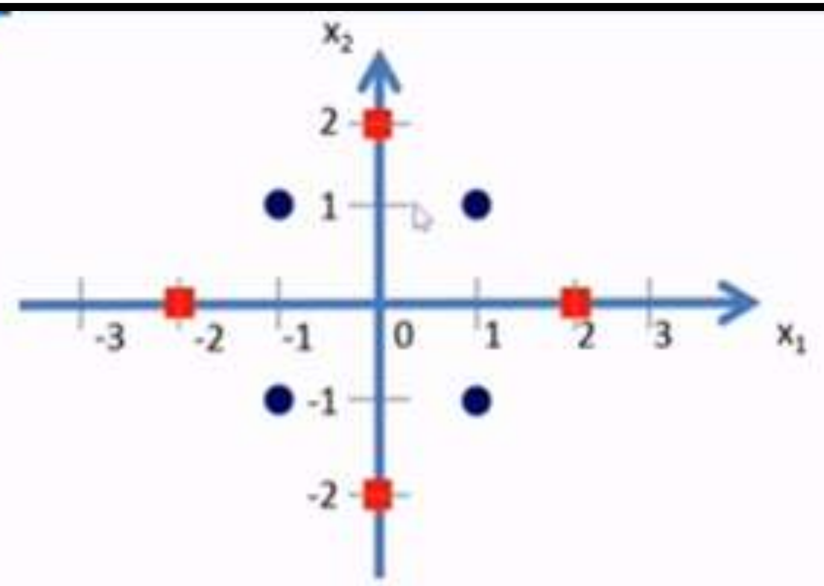
Suppose , If the Line is (0,1) ie, the line is parallel to X- Axis.

If the line is (1,1) , ie, the line is parallel to 45 Degrees.



EXAMPLE-2

In the **Given Dataset**, we **have 4 are positively labeled data sets and 4 are negatively labeled data sets.**



- Blue class vectors are: $\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}-1\\1\end{pmatrix}, \begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}1\\-1\end{pmatrix}$
- Red class vectors are: $\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}-2\\0\end{pmatrix}, \begin{pmatrix}0\\-2\end{pmatrix}$

Here our Goal is to separate Hyper plane that accurately separately two classes by using the Mapping Function RBF.

$$\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{cases}\begin{pmatrix}6-x_1+(x_1-x_2)^2\\6-x_2+(x_1-x_2)^2\end{pmatrix} & \text{if } \sqrt{x_1^2+x_2^2}\geq 2\\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise}\end{cases}$$

- Now let us transform the blue and red calss vectors using the non-linear mapping function Φ .

- $$\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases}\begin{pmatrix}6 - x_1 + (x_1 - x_2)^2\\6 - x_2 + (x_1 - x_2)^2\end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} \geq 2\\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise}\end{cases}$$

- Blue class vectors are: $\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}-1\\1\end{pmatrix}, \begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}1\\-1\end{pmatrix}$ no change since $\sqrt{x_1^2 + x_2^2} < 2$ for all the vectors

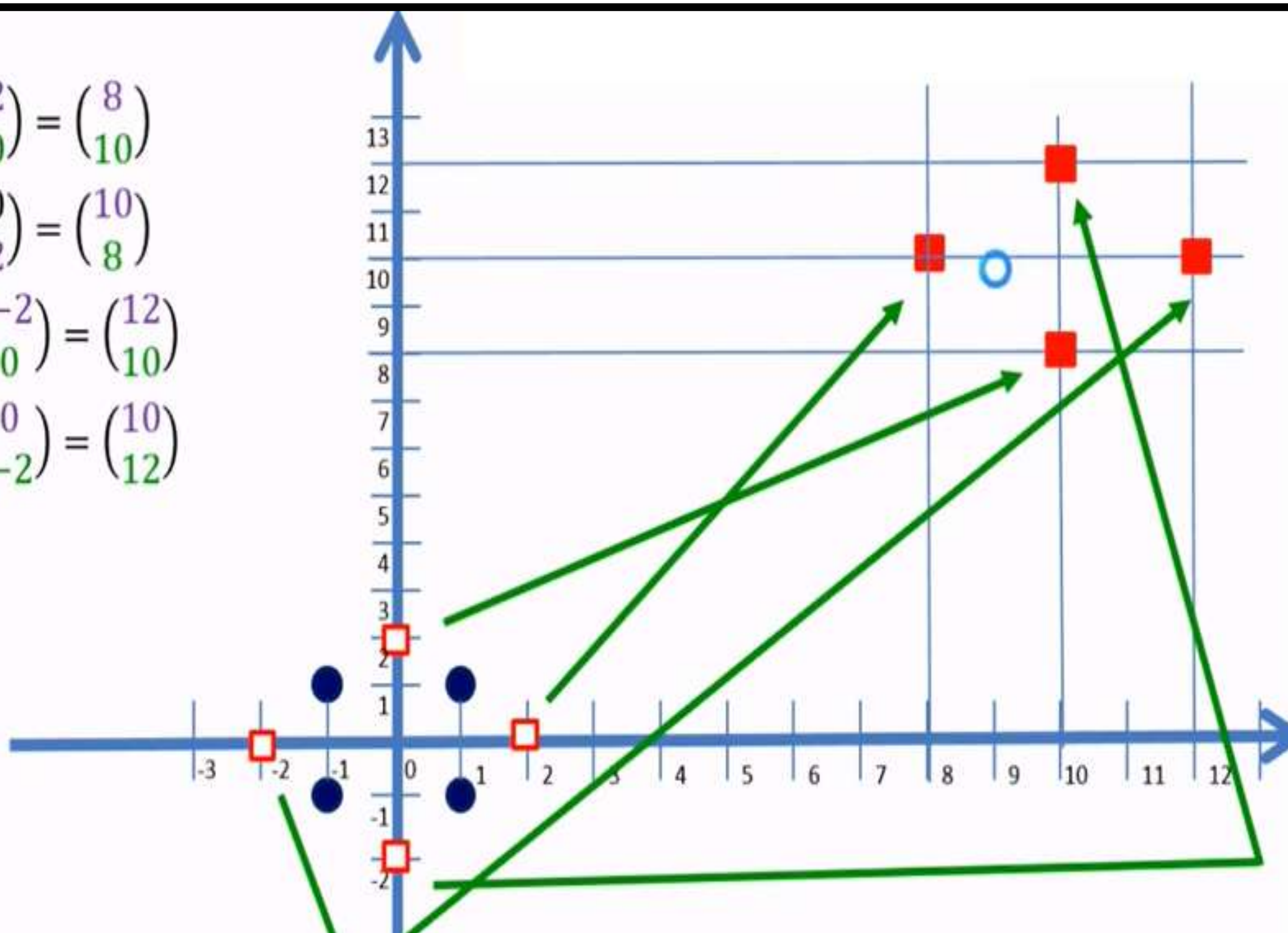
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{cases} \begin{pmatrix}6 - x_1 + (x_1 - x_2)^2\\6 - x_2 + (x_1 - x_2)^2\end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} \geq 2 \\ \begin{pmatrix}x_1\\x_2\end{pmatrix} & \text{otherwise} \end{cases}$

- Let us take Red class vectors : $\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}0\\2\end{pmatrix}, \begin{pmatrix}-2\\0\end{pmatrix}, \begin{pmatrix}0\\-2\end{pmatrix}$

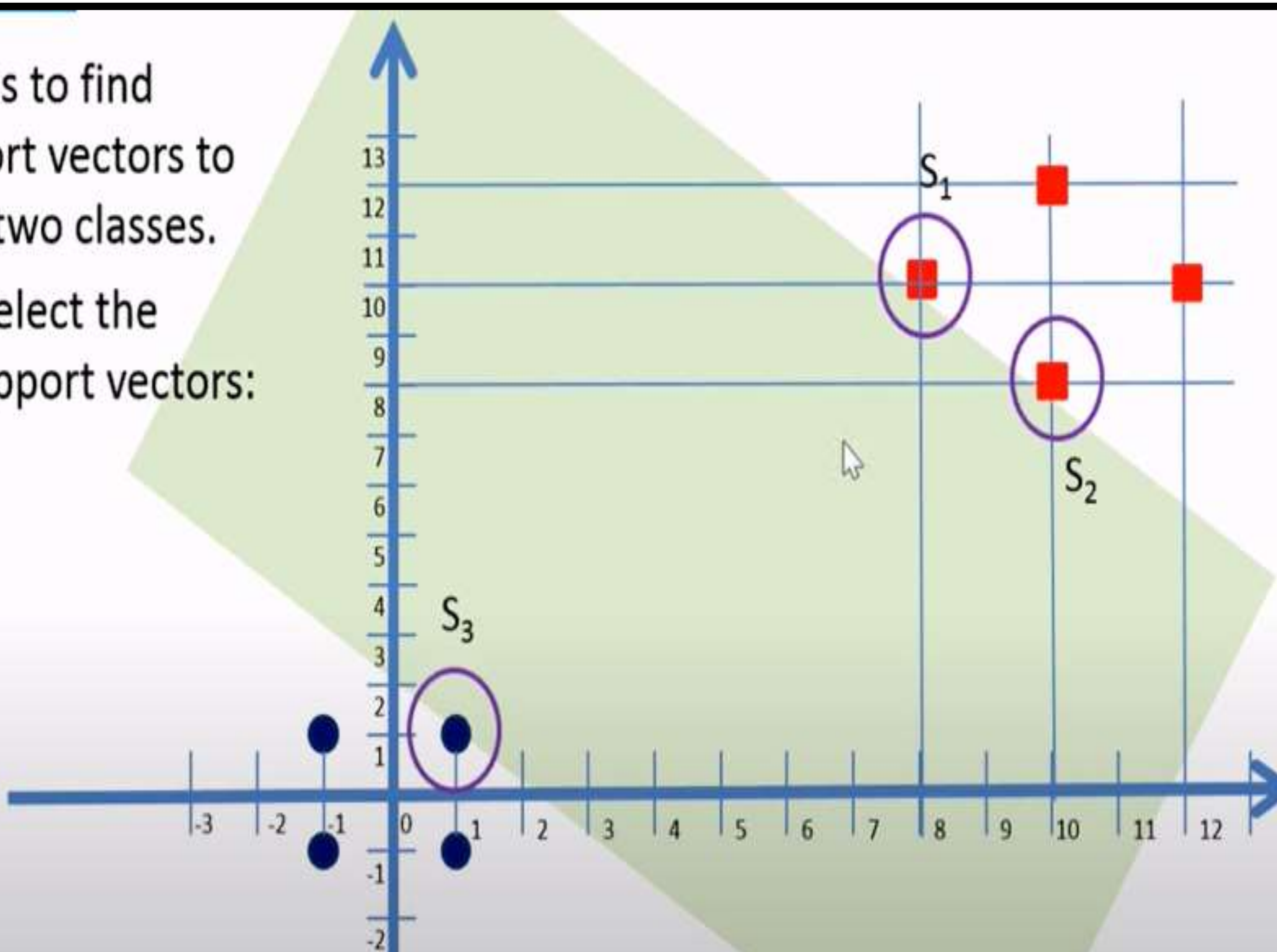
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}6 - 2 + (2 - 0)^2\\6 - 0 + (2 - 0)^2\end{pmatrix} = \begin{pmatrix}8\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\2\end{pmatrix} = \begin{pmatrix}6 - 0 + (0 - 2)^2\\6 - 2 + (0 - 2)^2\end{pmatrix} = \begin{pmatrix}10\\8\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}-2\\0\end{pmatrix} = \begin{pmatrix}6 + 2 + (-2 - 0)^2\\6 - 0 + (-2 - 0)^2\end{pmatrix} = \begin{pmatrix}12\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\-2\end{pmatrix} = \begin{pmatrix}6 - 0 + (0 + 2)^2\\6 + 2 + (0 + 2)^2\end{pmatrix} = \begin{pmatrix}10\\12\end{pmatrix}$

Draw the Hyper Plane by using updated Data Points

- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}2\\0\end{pmatrix} = \begin{pmatrix}8\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\2\end{pmatrix} = \begin{pmatrix}10\\8\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}-2\\0\end{pmatrix} = \begin{pmatrix}12\\10\end{pmatrix}$
- $\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \Phi\begin{pmatrix}0\\-2\end{pmatrix} = \begin{pmatrix}10\\12\end{pmatrix}$



- Now our task is to find suitable support vectors to classify these two classes.
- Here we will select the following 3 support vectors:
- $S_1 = \begin{pmatrix} 8 \\ 10 \end{pmatrix}$,
- $S_2 = \begin{pmatrix} 10 \\ 8 \end{pmatrix}$,
- and $S_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$



Here we will add the Bias value

$$S_1 = \begin{pmatrix} 8 \\ 10 \end{pmatrix} \qquad \widetilde{S}_1 = \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 10 \\ 8 \end{pmatrix} \qquad \widetilde{S}_2 = \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \widetilde{S}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\alpha_1 \widetilde{S_1}.\widetilde{S_1} + \alpha_2 \widetilde{S_2}.\widetilde{S_1} + \alpha_3 \widetilde{S_3}.\widetilde{S_1} = +1 \quad (+ve\ class)$$

$$\alpha_1 \widetilde{S_1}.\widetilde{S_2} + \alpha_2 \widetilde{S_2}.\widetilde{S_2} + \alpha_3 \widetilde{S_3}.\widetilde{S_2} = +1 \quad (+ve\ class)$$

$$\alpha_1 \widetilde{S_1}.\widetilde{S_3} + \alpha_2 \widetilde{S_2}.\widetilde{S_3} + \alpha_3 \widetilde{S_3}.\widetilde{S_3} = -1 \quad (-ve\ class)$$

- Let's substitute the values for $\widetilde{S_1}$, $\widetilde{S_2}$ and $\widetilde{S_3}$ in the above equations.

$$\widetilde{S_1} = \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} \quad \widetilde{S_2} = \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} \quad \widetilde{S_3} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} . \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} . \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} = +1$$

$$\alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} . \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} . \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} = +1$$

$$\alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} . \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} . \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1$$

After simplification we get:

$$165\,\alpha_1 + 161\,\alpha_2 + 19\,\alpha_3 = -1$$

$$161\,\alpha_1 + 165\,\alpha_2 + 19\,\alpha_3 = -1$$

$$19\,\alpha_1 + 19\,\alpha_2 + 3\,\alpha_3 = +1$$

Simplifying the above 3 simultaneous equations we get: α_1=α_2= 0.859 and α_3= -1.4219.

- The hyper plane that discriminates the positive class from the negative class is given by:

$$\tilde{w} = \sum_i \alpha_i \tilde{S}_i$$

- Substituting the values we get:

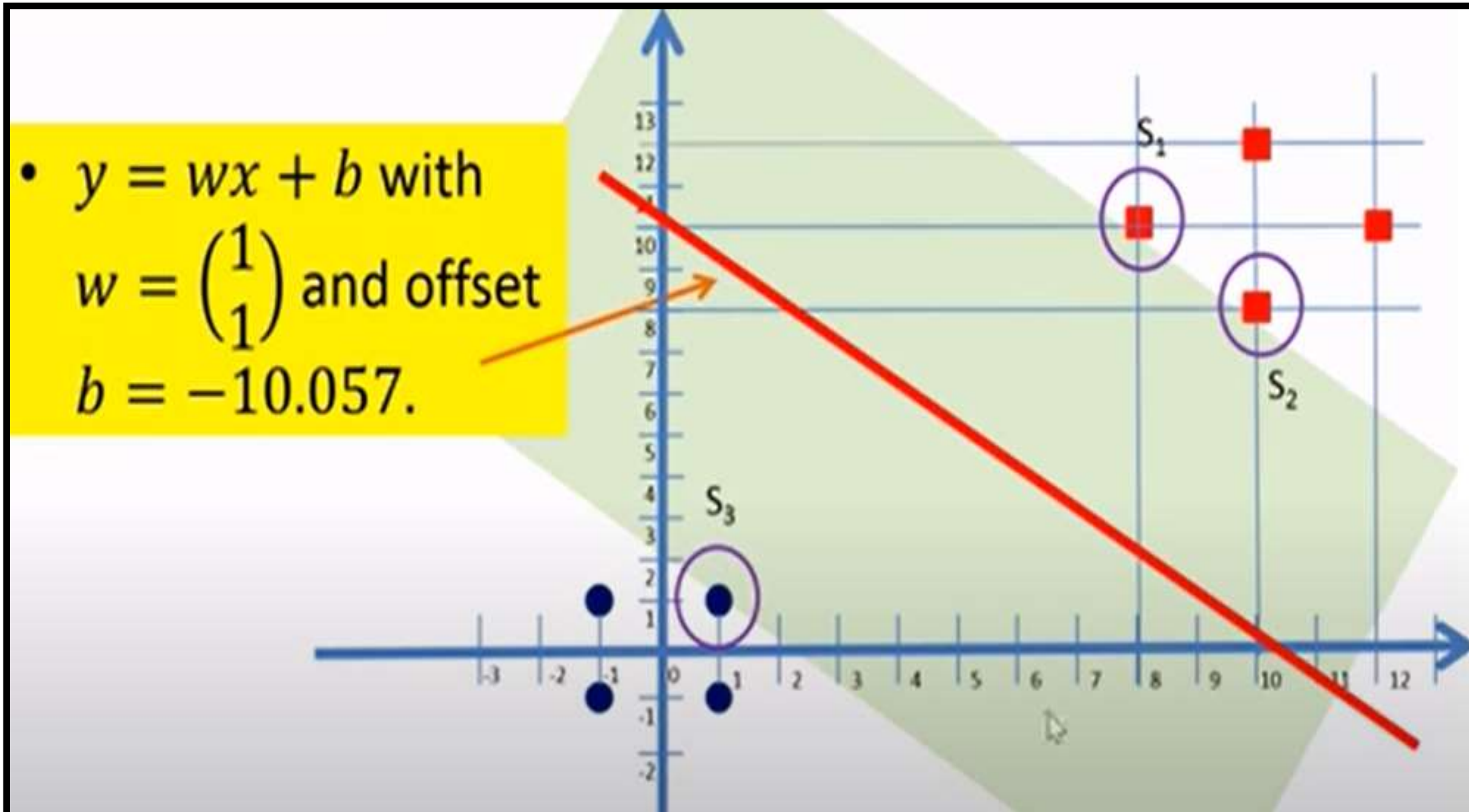
$$\tilde{w} = \alpha_1 \begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\tilde{w} = (0.0859).\begin{pmatrix} 8 \\ 10 \\ 1 \end{pmatrix} + (0.0859).\begin{pmatrix} 10 \\ 8 \\ 1 \end{pmatrix} + (-1.4219).\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.1243 \\ 0.1243 \\ -1.2501 \end{pmatrix}$$

- Our vectors are augmented with a bias.
- Hence we can equate the entry in $\tilde{w}$ as the hyper plane with an offset b.
- Therefore the separating hyper plane equation

$y = wx + b$ with $w = \begin{pmatrix} 0.125/0.125 \\ 0.125/0.125 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

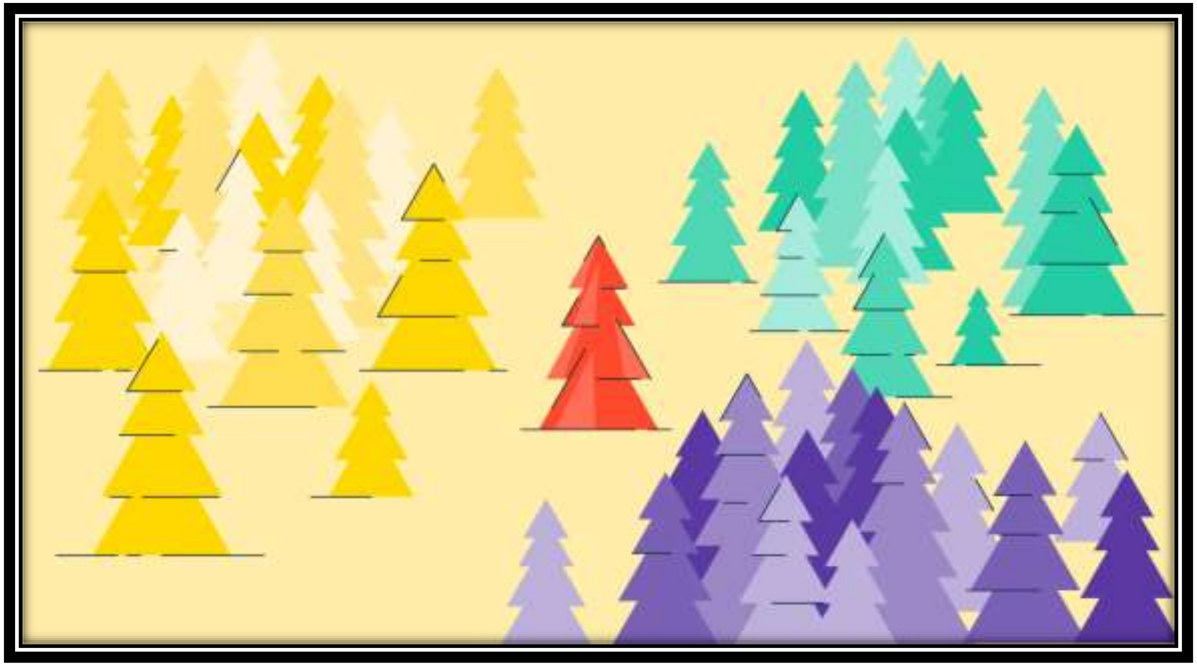
and an offset $b = -\frac{1.250}{0.125} = -10.057.$


• $y = wx + b$ with
$w = \begin{pmatrix}1\\1\end{pmatrix}$ and offset
$b = -10.057.$
S_1
S_2
S_3

THANK YOU

UNIT-IV

(Instance Based Learning)



Topics :

- K- Nearest Neighbor learning(KNN)
    - Choosing parameters for Nearest Neighbor
    - KNN Algorithm
    - Example of KNN

K-Nearest Neighbor(KNN) Learning

- The most basic **instance –based method** is the **K- Nearest Neighbor Algorithm.**
- **K-Nearest Neighbour** is one of the **simplest Machine Learning algorithm** based on **Supervised Learning technique.**
- It is used to **solve both classification and regression problems.** However, it's mainly used for **classification problems.**

CONTD..

- The K-nearest neighbors (KNN) algorithm is a **data classification method** for estimating the data point will become a **member of one group or another** based on what group the data points nearest to it belong to.
- The following **two properties** would **define KNN** well −
- **1. Lazy learning algorithm −** KNN is a lazy learning algorithm, because **it does not learn from the training set immediately** instead it stores the **dataset and at the time of classification**, it performs **an action on the dataset.**
- i.e, **instead of Learning** we have **only memorizing the data.**

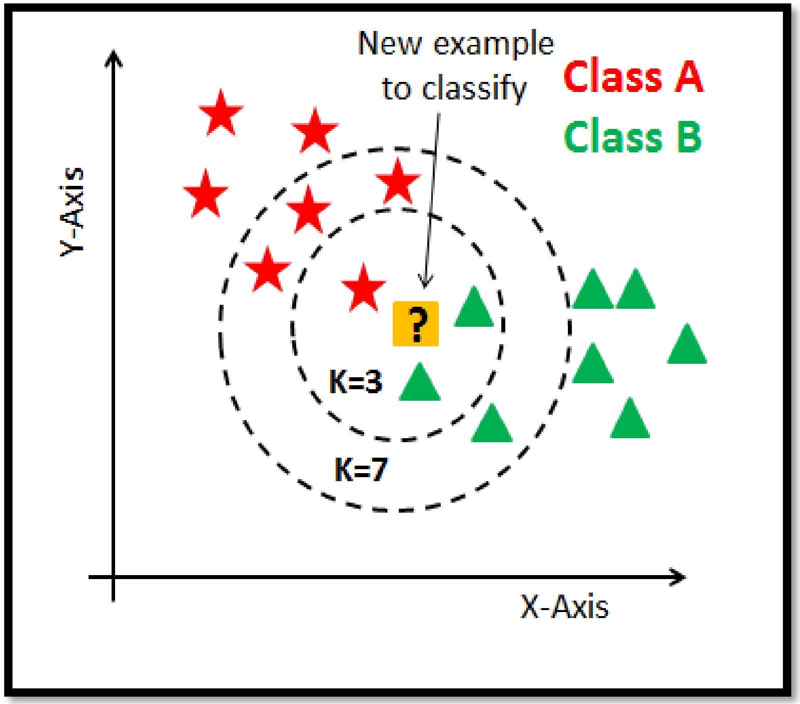
- **Eg: Learning and Memorizing**
- Suppose you have **preparing for the exam**, once **you write the exam** then what you have **learnt erase from your brain** that it is only **the memorizing.**
- But **instead of memorizing** , if **you learn and if you understand the concept** and if you **go to the exam**, **even after completing the exam also** you **will able to remember that topic.**
- **2. Non-parametric learning algorithm –** KNN is also a non-parametric learning algorithm because **it doesn't assume anything about the underlying data.**

KNN Example

- Suppose there are **two classes**, i.e., **Class A and Class B**, and we have a **new unknown data point "?"**, so this data point will lie in which of these classes.
- To **solve this problem**, we need a **K-NN algorithm**. With the help of K-NN, we can **easily identify the class of a particular dataset.**
- The data point is classified by a **majority vote of its neighbors**, with the data point being **assigned to the class.** Most commonly amongst its **K nearest neighbors measured by** a **Distance Function.**

CONTD..

Here, we can see that **if k = 3,** then based on the **distance function used**, the **nearest three neighbors of the data point is found** and based on the **majority votes of its neighbors,** the data point is **classified into a class**.



- In the case **of k = 3,** for the above diagram, it's **Class B**. Similarly, when **k = 7**, for the above diagram, based on the majority votes of its neighbors, the data point is classified to **Class A.**

1. Choosing Parameters for Nearest Neighbor

- Below are some points to remember while selecting the **value of K** in the **K-NN algorithm:**
- **1.** There is **no particular way to determine the best value for "K"**, so we need to try some values to **find the best out of them**. The most **preferred value for K is 5.**
- **2.** A **very low value for K** such as **K=1 or K=2**, can be **noisy and lead to the effects of outliers in the model.**
- **3. Large values for K are good**, but it **may find some difficulties**.

2. KNN ALGORITHM

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the **number K of the neighbors**

Step-2: Calculate the **Euclidean distance of K number of neighbors**

Step-3: Take the **K nearest neighbors** as per the **calculated Euclidean distance.**

Step-4: Among these k neighbors, **count the number of the data points in each category.**

Step-5: Assign the **new data points** to that category for which the **number of the neighbor is maximum.**

Step-6: **Our model is ready.**

CONTD..

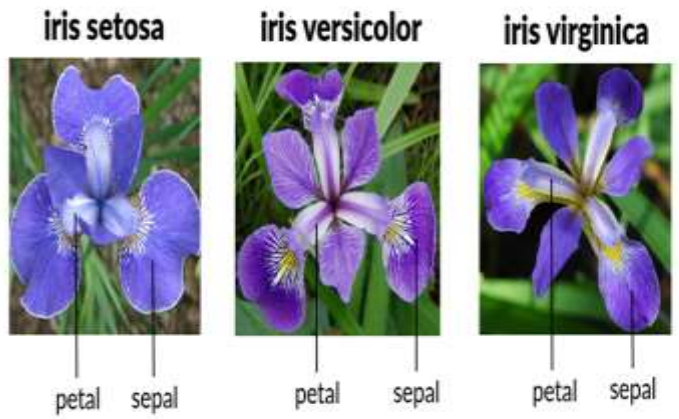
- There are **four ways to calculate the distance measure between** the **data point** and its **nearest neighbor:**
- **Euclidean distance,**
- **Manhattan distance,**
- **Hamming distance, and**
- **Minkowski distance**.
- **Out of the three, Euclidean distance** is the most **commonly used distance function or metric.**

$$\text{Euclidean Distance between } A_1 \text{ and } B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$$

3. Example of KNN

Example-1: Consider the given data set, it contains 15 examples.

Sepal Length	Sepal Width	Species
5.3	3.7	Setosa
5.1	3.8	Setosa
7.2	3.0	Virginica
5.4	3.4	Setosa
5.1	3.3	Setosa
5.4	3.9	Setosa
7.4	2.8	Virginica
6.1	2.8	Verscicolor
7.3	2.9	Virginica
6.0	2.7	Verscicolor
5.8	2.8	Virginica
6.3	2.3	Verscicolor
5.1	2.5	Verscicolor
6.3	2.5	Verscicolor
5.5	2.4	Verscicolor



- Here we have 2 features. i.e, sepal length and sepal width. And here the Target is Species.
- i.e, Target has 3 possibilities : Setosa, Virginica, Versicolor.
- Now given the Value of K- we want to classify the new example. ie,

Sepal Length	Sepal Width	Species
5.2	3.1	?

- Suppose the value of K- may be 1 , 2 ,3 soon.
- Here how to classify the species with the help of K-Nearest Neighbor Algorithm.

Step-1: First we need to **calculate the Distance.** Ie, here we have to use the **Euclidean Distance.**

Sepal Length	Sepal Width	Species
5.3	3.7	Setosa
5.1	3.8	Setosa
7.2	3.0	Virginica
5.4	3.4	Setosa
5.1	3.3	Setosa
5.4	3.9	Setosa
7.4	2.8	Virginica
6.1	2.8	Verscicolor
7.3	2.9	Virginica
6.0	2.7	Verscicolor
5.8	2.8	Virginica
6.3	2.3	Verscicolor
5.1	2.5	Verscicolor
6.3	2.5	Verscicolor
5.5	2.4	Verscicolor

Step 1: Find Distance

$$\text{Distance (Sepal Length, Sepal Width)} = \sqrt{(x-a)^2 + (y-b)^2}$$

$$\text{Distance (Sepal Length, Sepal Width)} = \sqrt{(5.2-5.3)^2 + (3.1-3.7)^2}$$

$$\text{Distance (Sepal Length, Sepal Width)} = 0.608$$

Sepal Length	Sepal Width	Species	Distance
5.3	3.7	Setosa	0.608

Sepal Length	Sepal Width	Species
5.2	3.1	?

In the same way we have to calculate all the examples. And finally will get

Sepal Length	Sepal Width	Species	Distance
5.3	3.7	Setosa	0.608
5.1	3.8	Setosa	0.707
7.2	3.0	Virginica	2.002
5.4	3.4	Setosa	0.36
5.1	3.3	Setosa	0.22
5.4	3.9	Setosa	0.82
7.4	2.8	Virginica	2.22
6.1	2.8	Verscicolor	0.94
7.3	2.9	Virginica	2.1
6.0	2.7	Verscicolor	0.89
5.8	2.8	Virginica	0.67
6.3	2.3	Verscicolor	1.36
5.1	2.5	Verscicolor	0.60
6.3	2.5	Verscicolor	1.25
5.5	2.4	Verscicolor	0.75

Step-2: To find Rank. Ie, here the **distance** which is having **minimum is the First Rank.**

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

Step-3: To find K-Nearest Neighbour. Here, we have to take k=1 rank example etc.

Sepal Length	Sepal Width	Species	Distance	Rank
5.3	3.7	Setosa	0.608	3
5.1	3.8	Setosa	0.707	6
7.2	3.0	Virginica	2.002	13
5.4	3.4	Setosa	0.36	2
5.1	3.3	Setosa	0.22	1
5.4	3.9	Setosa	0.82	8
7.4	2.8	Virginica	2.22	15
6.1	2.8	Verscicolor	0.94	10
7.3	2.9	Virginica	2.1	14
6.0	2.7	Verscicolor	0.89	9
5.8	2.8	Virginica	0.67	5
6.3	2.3	Verscicolor	1.36	12
5.1	2.5	Verscicolor	0.60	4
6.3	2.5	Verscicolor	1.25	11
5.5	2.4	Verscicolor	0.75	7

Step 3: Find the Nearest Neighbor

If k = 1 – Setosa

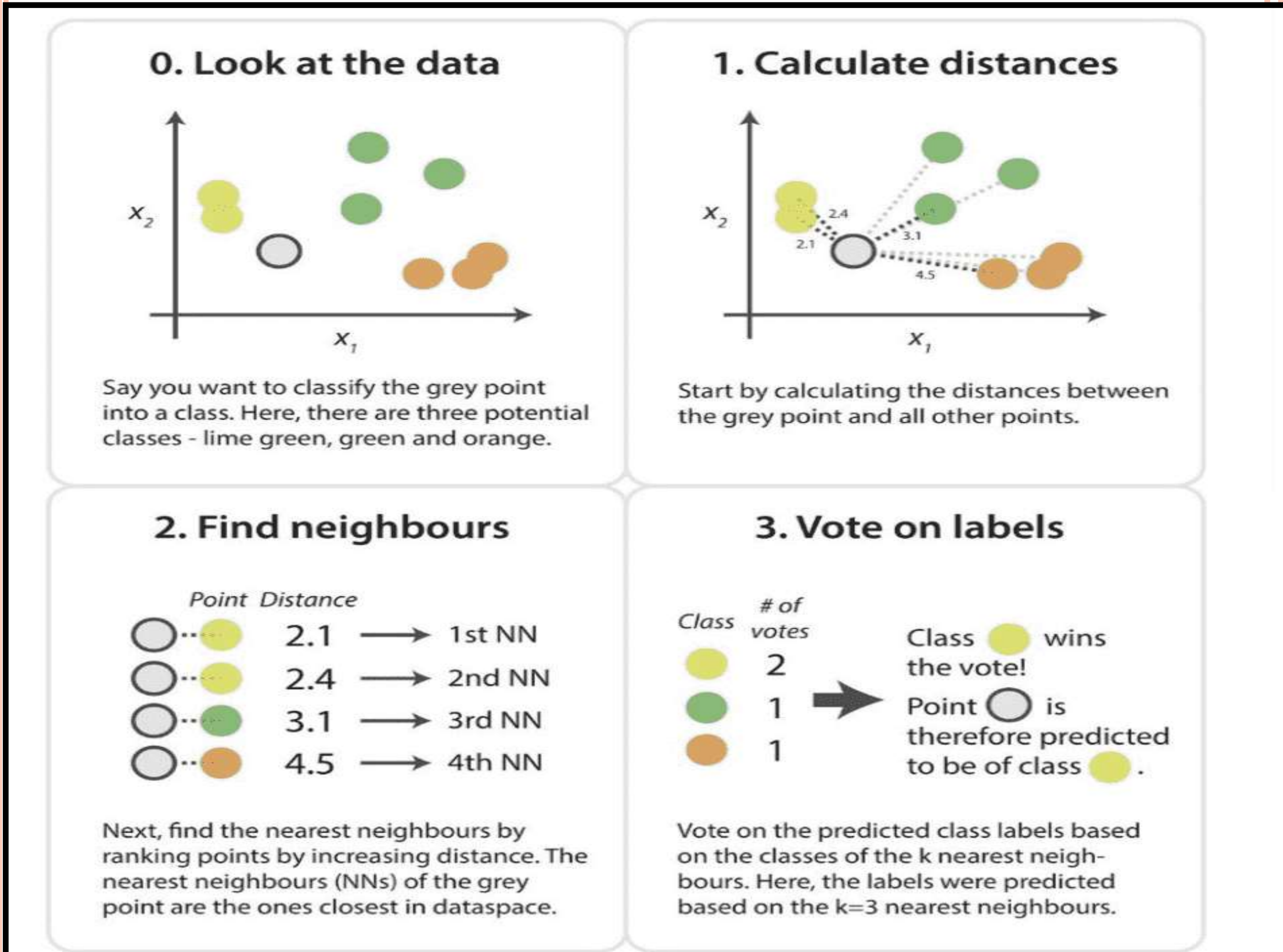
If k = 2 – Setosa

If k = 5 – Setosa

- Here, we can observe that , suppose we have to take **k=1 is Setosa**, **K=2 Setosa,** **K=3 Setosa**, **K=4 Versicolor**, but majority of voting is Setosa. So Answer is Setosa for K=4.

K=5 Virginica, but **majority of voting is Setosa**. So Answer is **Setosa for K=5** .

- **So Our new Example goes to Setosa.**


0. Look at the data
x_2
x_1
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.
1. Calculate distances
x_2
2.4
2.1
3.1
4.5
x_1
Start by calculating the distances between the grey point and all other points.
2. Find neighbours
Point Distance
2.1 → 1st NN
2.4 → 2nd NN
3.1 → 3rd NN
4.5 → 4th NN
Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.
3. Vote on labels
Class
# of votes
2
1
1
Class wins the vote!
Point is therefore predicted to be of class .
Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

EXAMPLE-2

To predict the Class Label for new instance.

Height (CM)	Weight (KG)	Class
167	51	Underweight
182	62	Normal
176	69	Normal
173	64	Normal
172	65	Normal
174	56	Underweight
169	58	Normal
173	57	Normal
170	55	Normal
170	57	?

➢ 1. First we can calculate the Distance Formula:

THE DISTANCE FORMULA

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Height (CM)	Weight (KG)	Class	Distance
167	51	Underweight	6.7
182	62	Normal	13
176	69	Normal	13.4
173	64	Normal	7.6
172	65	Normal	8.2
174	56	Underweight	4.1
169	58	Normal	1.4
173	57	Normal	3
170	55	Normal	2
170	57	?	

➢ 2. next, we can calculate the Rank:

Height (CM)	Weight (KG)	Class	Distance	Rank
169	58	Normal	1.4	1
170	55	Normal	2	2
173	57	Normal	3	3
174	56	Underweight	4.1	4
167	51	Underweight	6.7	5
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

➢ 3. next, we can find the K-nearest Neighbor:

Height (CM)	Weight (KG)	Class	Distance	Rank
169	58	Normal ✓	1.4	1 ✓
170	55	Normal ✓	2	2 ✓
173	57	Normal ✓	3	3 ✓
174	56	Underweight ✓	4.1	4 ✓
167	51	Underweight ✓	6.7	5 ✓
173	64	Normal	7.6	6
172	65	Normal	8.2	7
182	62	Normal	13	8
176	69	Normal	13.4	9
170	57	?		

- If K=1, Normal
- If K=2, Normal
- If K=3, Normal
- If K=4, Normal
- If K=5, Normal

Based on the K- Neighbors, our new Instance is also Normal Class.

Example-3

BMI	Age	Sugar
33.6	50	1
26.6	30	0
23.4	40	0
43.1	67	0
35.3	23	1
35.9	67	1
36.7	45	1
25.7	46	0
23.3	29	0
31	56	1

- **Apply** K nearest neighbor classifier to predict the diabetic patient with the given features BMI, Age. If the training examples are,
- Assume K=3,
- Test Example BMI=43.6, Age=40, Sugar=?

CONTD..

BMI	Age	Sugar
33.6	50	1
26.6	30	0
23.4	40	0
43.1	67	0
35.3	23	1
35.9	67	1
36.7	45	1
25.7	46	0
23.3	29	0
31	56	1

- First Calculate the distance between the test instance and training instances.

Test Example

BMI=43.6, Age=40, Sugar=?

- *Distance* $= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Contd..

BMI	Age	Sugar	Distance	
33.6	50	1	$\sqrt{(43.6-33.6)^2+(40-50)^2}$	14.14
26.6	30	0	$\sqrt{(43.6-26.6)^2+(40-30)^2}$	19.72
23.4	40	0	$\sqrt{(43.6-23.4)^2+(40-40)^2}$	20.20
43.1	67	0	$\sqrt{(43.6-43.1)^2+(40-67)^2}$	27.00
35.3	23	1	$\sqrt{(43.6-35.3)^2+(40-23)^2}$	18.92
35.9	67	1	$\sqrt{(43.6-35.9)^2+(40-67)^2}$	28.08
36.7	45	1	$\sqrt{(43.6-36.7)^2+(40-45)^2}$	8.52
25.7	46	0	$\sqrt{(43.6-25.7)^2+(40-46)^2}$	18.88
23.3	29	0	$\sqrt{(43.6-23.3)^2+(40-29)^2}$	23.09
31	56	1	$\sqrt{(43.6-31)^2+(40-56)^2}$	20.37

Test Example

BMI=43.6, Age=40, Sugar=?

CONTD..

BMI	Age	Sugar	Distance	Rank
33.6	50	1	14.14	2
26.6	30	0	19.72	
23.4	40	0	20.20	
43.1	67	0	27.00	
35.3	23	1	18.92	
35.9	67	1	28.08	
36.7	45	1	8.52	1
25.7	46	0	18.88	3
23.3	29	0	23.09	
31	56	1	20.37	

Test Example

BMI=43.6, Age=40, Sugar=?

Sugar = 1

EXAMPLE-4

- Given the following training instances (see table), each having two attributes (x1 and x2). Compute the class label for test instance t1 = (3,7) using three-nearest neighbors (k=3).

Training Instance	x_1	x_2	Output
I_1	7	7	0
I_2	7	4	0
I_3	3	4	1
I_4	1	4	1

➢ First we can calculate Distance, Rank

- Computing three nearest-neighbors for test instance

  t1 = (3,7) using Euclidean distance

Training Instance	x_1	x_2	Output	Distance	Neighbor Rank
I_1	7	7	0	$\sqrt{(7-3)^2+(7-7)^2}=4$	3
I_2	7	4	0	$\sqrt{(7-3)^2+(4-7)^2}=5$	4
I_3	3	4	1	$\sqrt{(3-3)^2+(4-7)^2}=3$	1
I_4	1	4	1	$\sqrt{(1-3)^2+(4-7)^2}$ $=3.6$	2

- **Next, we can find the Nearest Neighbor is K=3.**
- **So we can find first three K- nearest Neighbors.ie, I3, I4, I1**
- Here we can **observe that the Majority** . So here the **majority in this case is "1".**

t1 = (3,7) using Euclidean distance K=3 t1 = (3,7) → 1 ✓

Applications of KNN

1. **Finance –** financial institutes will predict the credit rating of customers.

2. **Healthcare** – gene expression.

3. **Political Science –** classifying potential voters in two classes will vote or won't vote.

4. **Handwriting detection.**

5. **Image Recognition.**

6. **Video Recognition.**

7. **Pattern Recognition**

text mining


agriculture


financial


healthcare

Weighted Nearest neighbour Algorithm

- Given the following training instances (see table), each having two attributes (x1 and x2). Compute the class label for test instance t1 = (3,7) using three-nearest neighbors (k=3).

Training Instance	x_1	x_2	Output
I_1	7	7	0
I_2	7	4	0
I_3	3	4	1
I_4	1	4	1

➤ First we can calculate Distance, Rank

- Computing three nearest-neighbors for test instance

t1 = (3,7) using Euclidean distance

Training Instance	x_1	x_2	Output	Distance	Neighbor Rank
I_1	7	7	0	$\sqrt{(7-3)^2+(7-7)^2}=4$	3
I_2	7	4	0	$\sqrt{(7-3)^2+(4-7)^2}=5$	4
I_3	3	4	1	$\sqrt{(3-3)^2+(4-7)^2}=3$	1
I_4	1	4	1	$\sqrt{(1-3)^2+(4-7)^2}$ $=3.6$	2

- Next, We can Calculated as **Inverse Square Distance**. Ie, we can considered as the **Weight for that particular instance.**
- **That is also called as " Weighted Nearest Neighbor Algorithm"**

Training Instance	x_1	x_2	Output	Distance (d)	d^2	Vote	Rank
I_1	7	7	0	4	16	1/16 = 0.06	3
I_2	7	4	0	5	25	1/25 = 0.04	4
I_3	3	4	1	3	9	1/9 = 0.11	1
I_4	1	4	1	3.6	12.96	1/12.96 = 0.08	2

- Here , we observe that **"Minimum Distance and Maximum Weight"**
- **Next, we can find the Nearest Neighbor is K=3.**
- **So we can find first three K- nearest Neighbors. ie, I3, I4, I1**
- Here we can **observe that the Majority** . So here the **majority in this case is "1".**

t1 = (3,7) using Euclidean distance K=3 t1 = (3,7) → 1 ✓

THANK YOU

UNIT-IV
(CASE BASED REASONING)

Topics :

- Case Based Reasoning
  - CBR Workflow
  - CBR Example
  - Real Time Scenario

Case Based Reasoning

- All **instance based learners** have **3 properties.**
  - **1.** They are **Lazy Learners**
  - **2. Classification is different for each instance**
  - 3. Instances are represented with **n-dimensional Euclidean Space.**
- CBR is any kind of **problem-solving approach** that uses **past solutions(experiences) to solve similar problems.**
- **Ie, In CBR**, **everything is considered** as **case** and **based on previous cases** , then **we can find the solution.**
- Case-based reasoning is **all around us.**

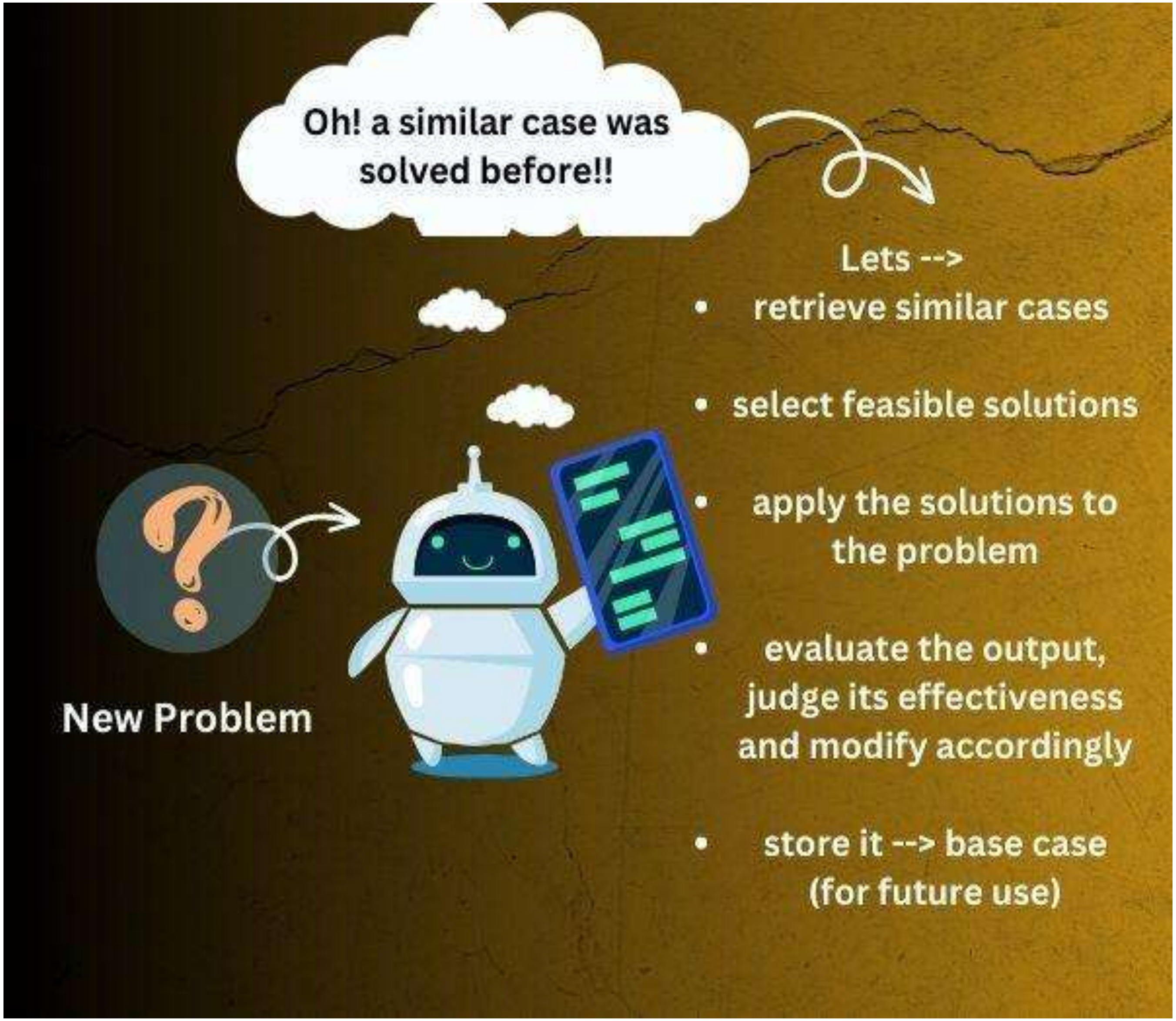
CONTD..

- **For example, Google Maps** uses **case-based reasoning** to tell you **how long your journey** will take by **examining the patterns of past users** to see **how long it took them to get from point A to point B**.
- Even **if your path is from two slightly different points**, it makes **inferences on how long your journey** will take.
- Generally, In **every Machine Learning Algorithm** we have **instances.**
- Here in **case of CBR instances are represented** as **Symbols not values.**

Definition of CBR

- **Definition:** **Case-based reasoning**, **a new problem** is **solved by adapting solutions** that were **also useful in the past**.
- Therefore, it is also referred to as an **experience-based approach**/ **intelligent problem-solving method**.
- Therefore, it means **learning from past experiences** and using that .

Definition of CBR

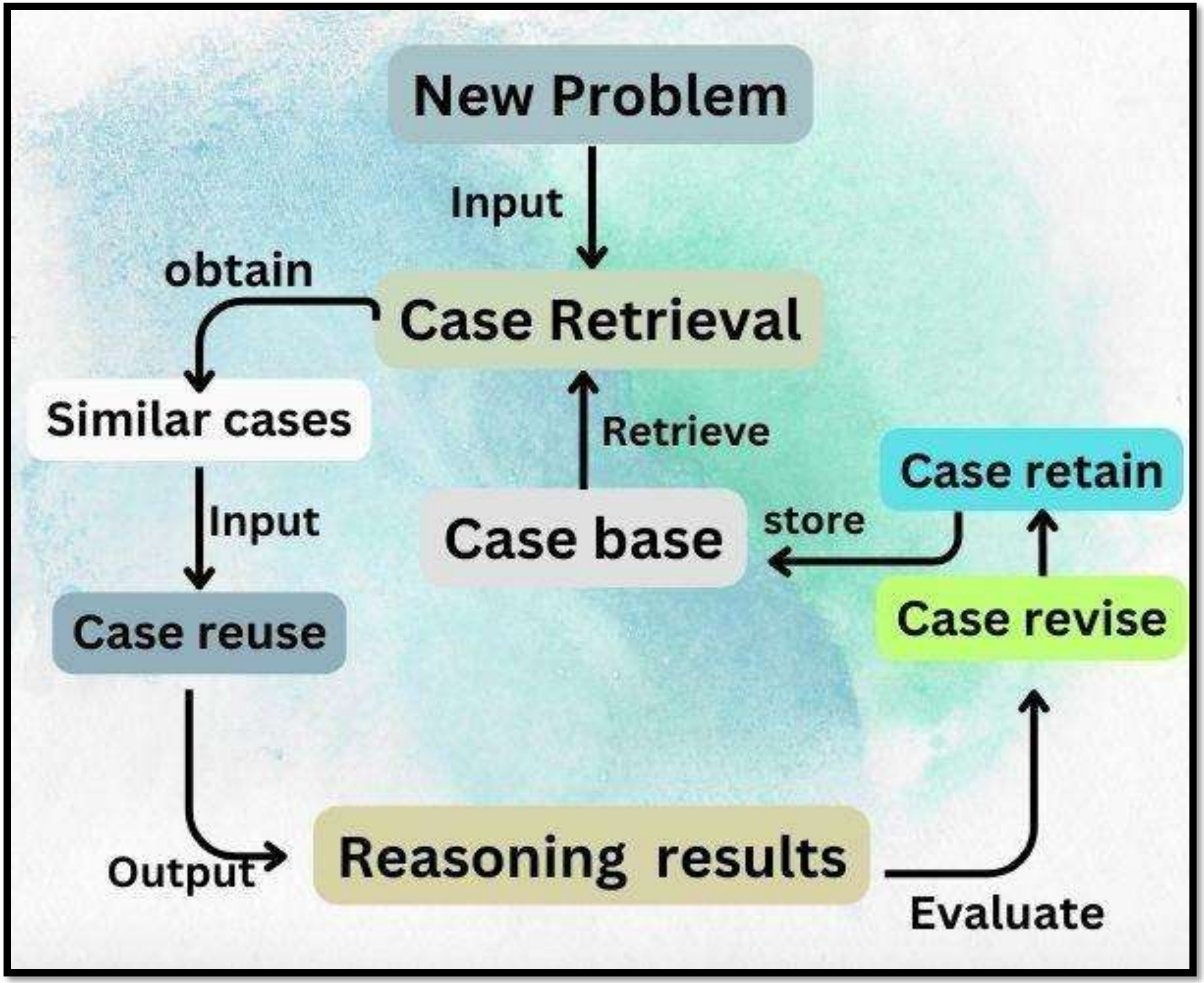


- **Case-Based Reasoning classifiers (CBR)** use a **database of problem solutions** to **solve new problems.**
- **CBR methods can be coupled with other methods**
  - **1.** Artificial Neural Networks
  - **2.** Classification
  - **3.** Optimization Algorithm
  - **4.** Genetic Algorithm
  - **5.** Fuzzy Logic etc.

CBR Life Cycle

- The CBR (Case-Based Learning) cycle is an **iterative process** that tells us how **a new problem is approached**.
- It refers to **collecting together past experiences** and making **use of relevant information**.
- The main steps in the Case-based reasoning cycle.

CBR Life Cycle



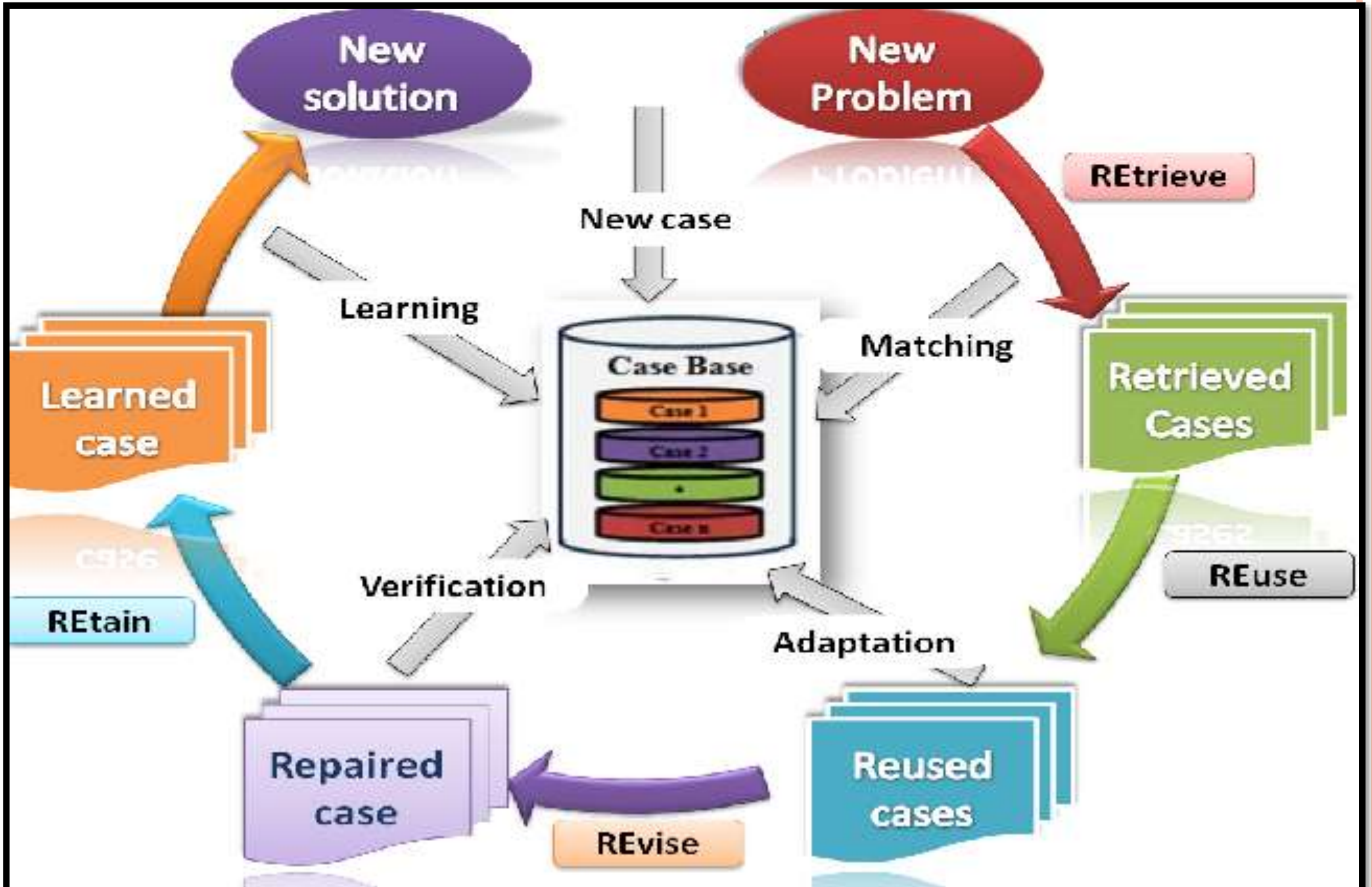
CONTD..

- **New Problem:** The CBR cycle starts when a **new problem arrives.**
- **Case retrieval:** After properly **analyzing the problem**, the **relevant information is extracted** by **comparing similar cases with the newly arrived problem**. Therefore, we put together useful similar cases to solve the problem.
- **Case reuse:** As a next step in the cycle, the **past information is reused,** and **feasible solutions are selected from similar cases**.

- **Reasoning results:** It refers to applying the **filtered information and solutions** by **analyzing previous similar cases to the present problem.** In this step, we generally use **algorithms and heuristics** to narrow down to a solution.
- **Case revision:** After applying the necessary information, the next step is to evaluate the **output→judge its effectiveness** and **produce feedback**.
  - Suppose the **result's quality is not up to the mark**. In that case, the **solution is modified according to the feedback,** and an **efficient solution is recorded for future similar cases like this.**

<u>Case retain:</u> **Storing this new problem-solving method in the memory system**

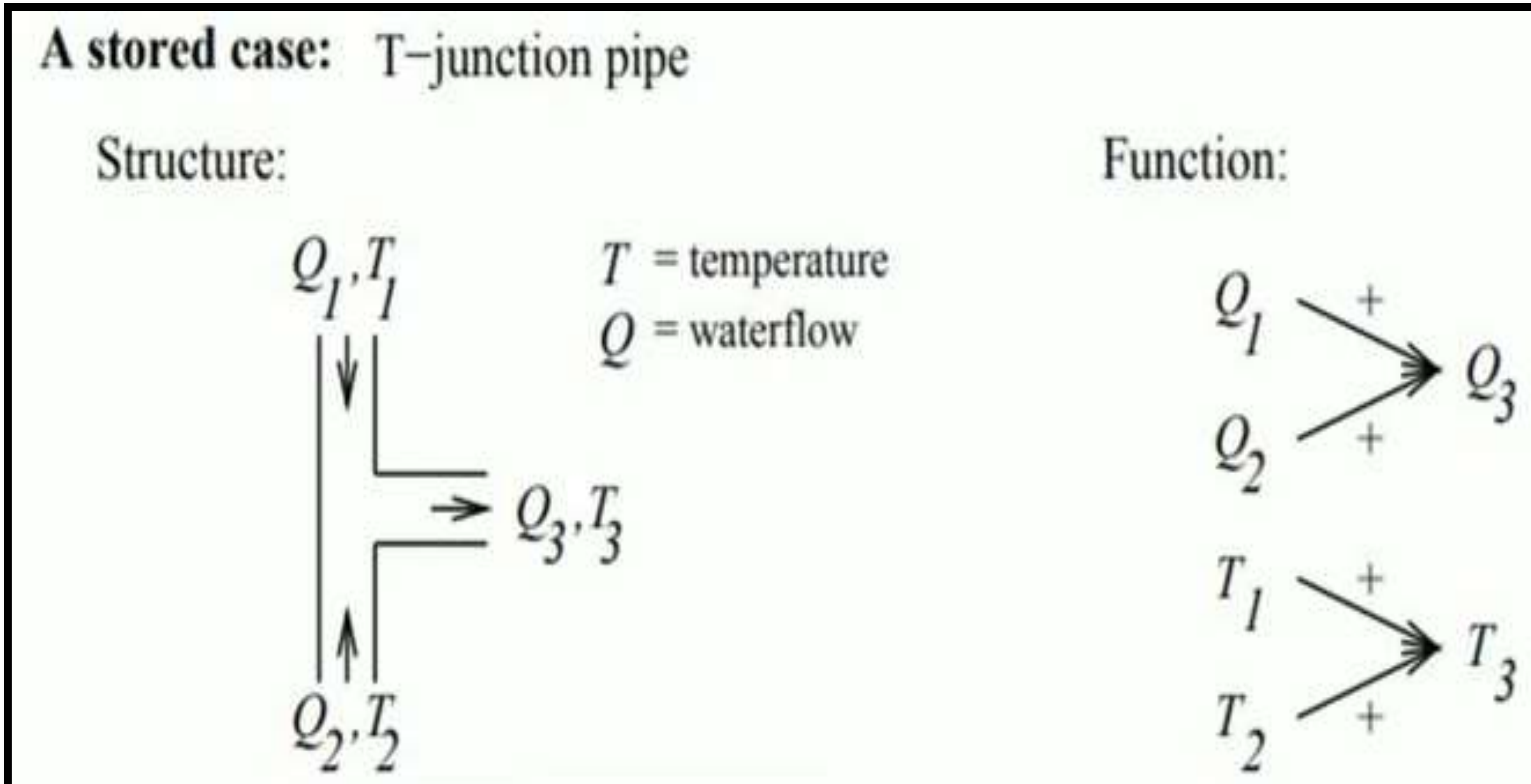
CBR Workflow



Prototypical Example of Case Based Reasoning

- **A CADET** System Employs **Case Based Reasoning** to assist in the **Conceptual Design** of simple **mechanical devices** such as **Water Faucets**.
- Here It uses a **Library Containing Approximately 75 Previous Designs** to suggest conceptual Designs to meet specifications of **new Design Problems**
- Each **Instance stored a Memory (Eg: Water Pipe)** is represented by **describing both its structure and Qualitative Function**

- **In our homes, now a days modern based taps are worked.**
- **Here we have to take T- Junction Tap.**
- Ie, **we lift the tap** we will be **getting the water.** And based on the **direction of the tap , ie.** if it is **completely towards left side** , we will **get hot water. Etc.**
- **In our example,** we can represented with the help of **symbols not values.**

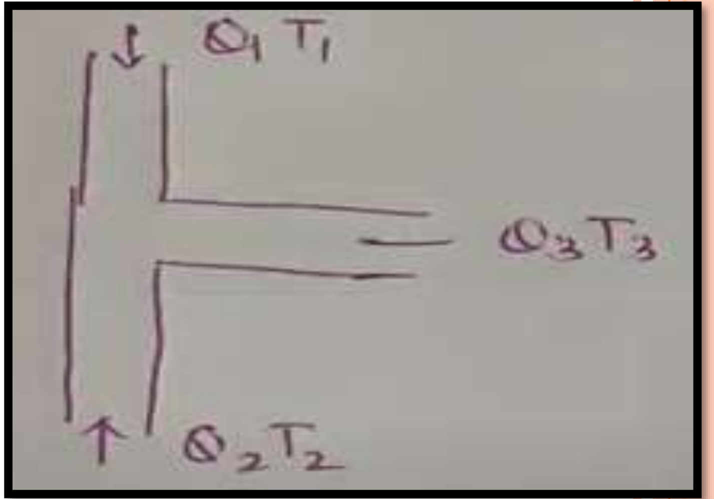

A stored case: T−junction pipe
Structure:
Function:
Q_1, T_1
T = temperature
Q = waterflow
Q_3, T_3
Q_2, T_2
Q_1
Q_2
Q_3
T_1
T_2
T_3
+
+
+
+

- In **this figure,**

Q- Means **Water flow** and

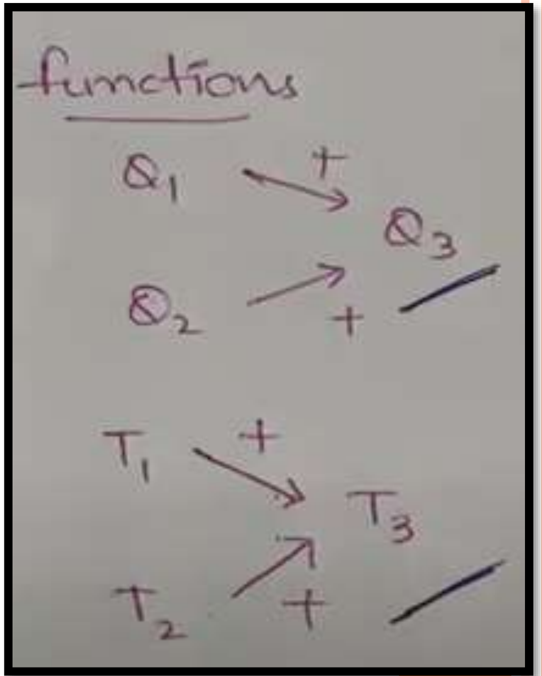
T- Means **Temperature**.

- so here in this figure, **Q1T1 & Q2T2** and **Q3T3** is the **combo of both.**



- **here we used the functions Q1 and Q2 will give you Q3**

And T1 and T2 will give you T3.



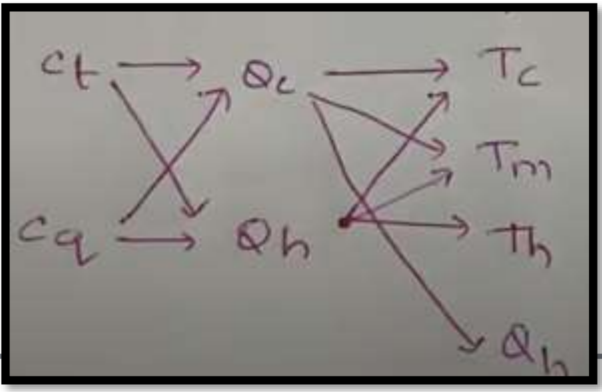
- So this is the **basic system** we have **in our house**.
- These **type of examples** are defined in **CADET System.**

CONTD..

- Suppose **we need to construct** and **manufacture of another tap.** Which will **control the temperature** and **water flow.**
- So here we have **only the option to choose** the **Hot Water** and **Cold Water.** But we **don't have the option** to choose the **Flow of water** or we **don't have an option** to choose **the Temperature.**
- but here in our system **, we can control the Temperature.**
- **SO, here what we can do?**

CONTD..

- So, here we have to take the **help of the system** based on **existing system** and we will do **some modifications** in the **existing system** in order to **get the system that we have desired**.
- _**Here,** **C_t stands for Control of Temperature and C_q stands for Control of Water Flow**
- **Q_c and Q_h - is the cold water and Hot water** and **Temperatures** are **various parameters etc.**



CBR Vs other Techniques

- The case-based reasoning is an exceptionally well-designed technique that uses past experiences to improve its performance.

1. **Rule-based systems:** In rule-based reasoning, **pre-defined rules are used for solving problems**. **Experts in the field design this set of rules**.

   On the other hand, **CBR is efficient in handling situations** where the **rules may not be efficient** or are not present.

Real time Example: An application of CBR with ML for forensic Autopsy

- Forensic relevant experts need to analyze collection of evidence.
- They need to interpret using own theory.
- Theories plus experiences can support the analysis of digital evidence.
- Shortages of human resources and time.
- Therefore, machine intelligence comes in useful.

- **System Flow graph**
- **Step-1:**

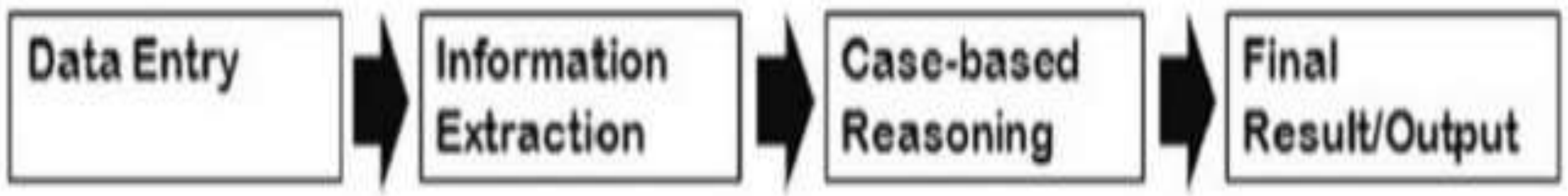
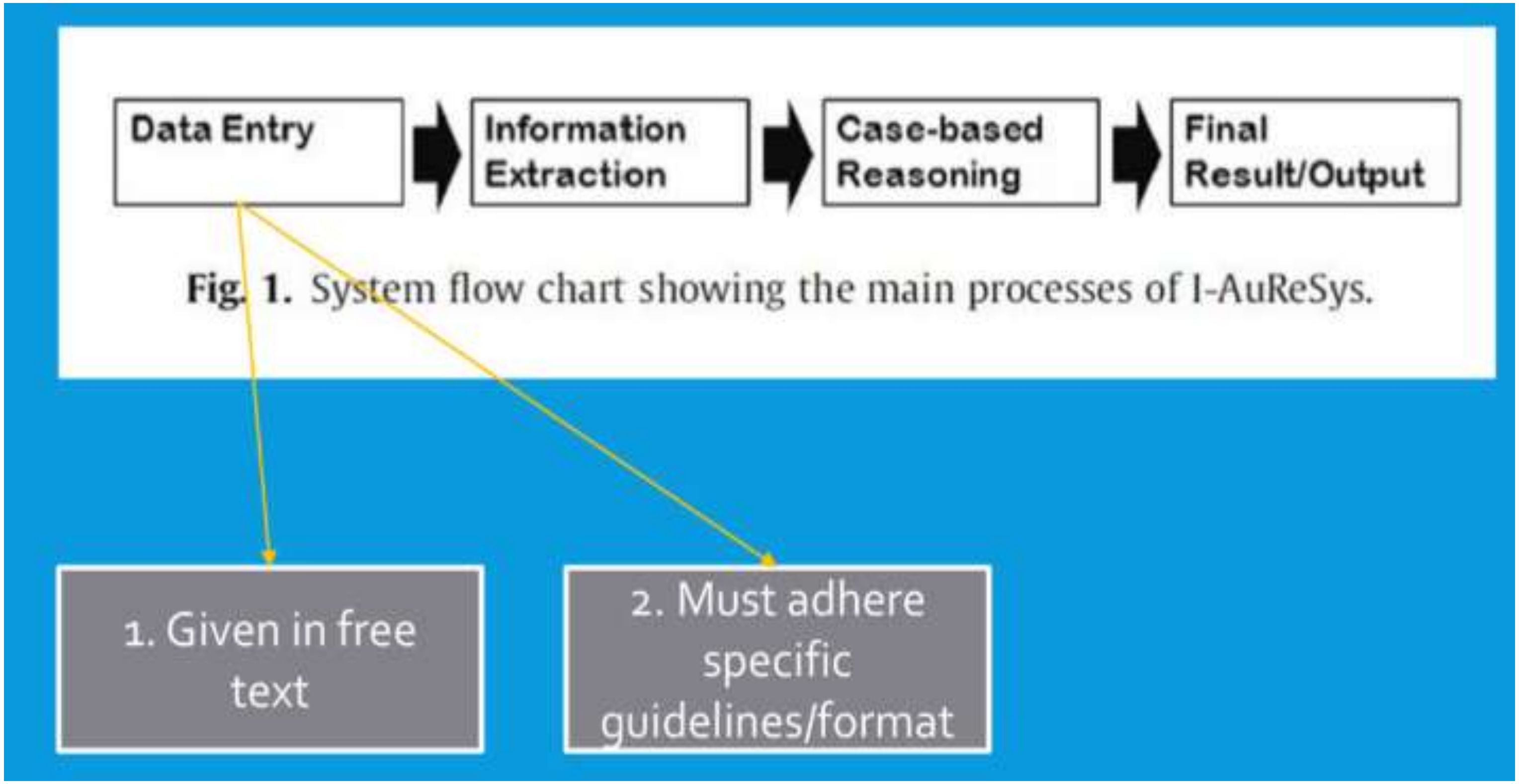


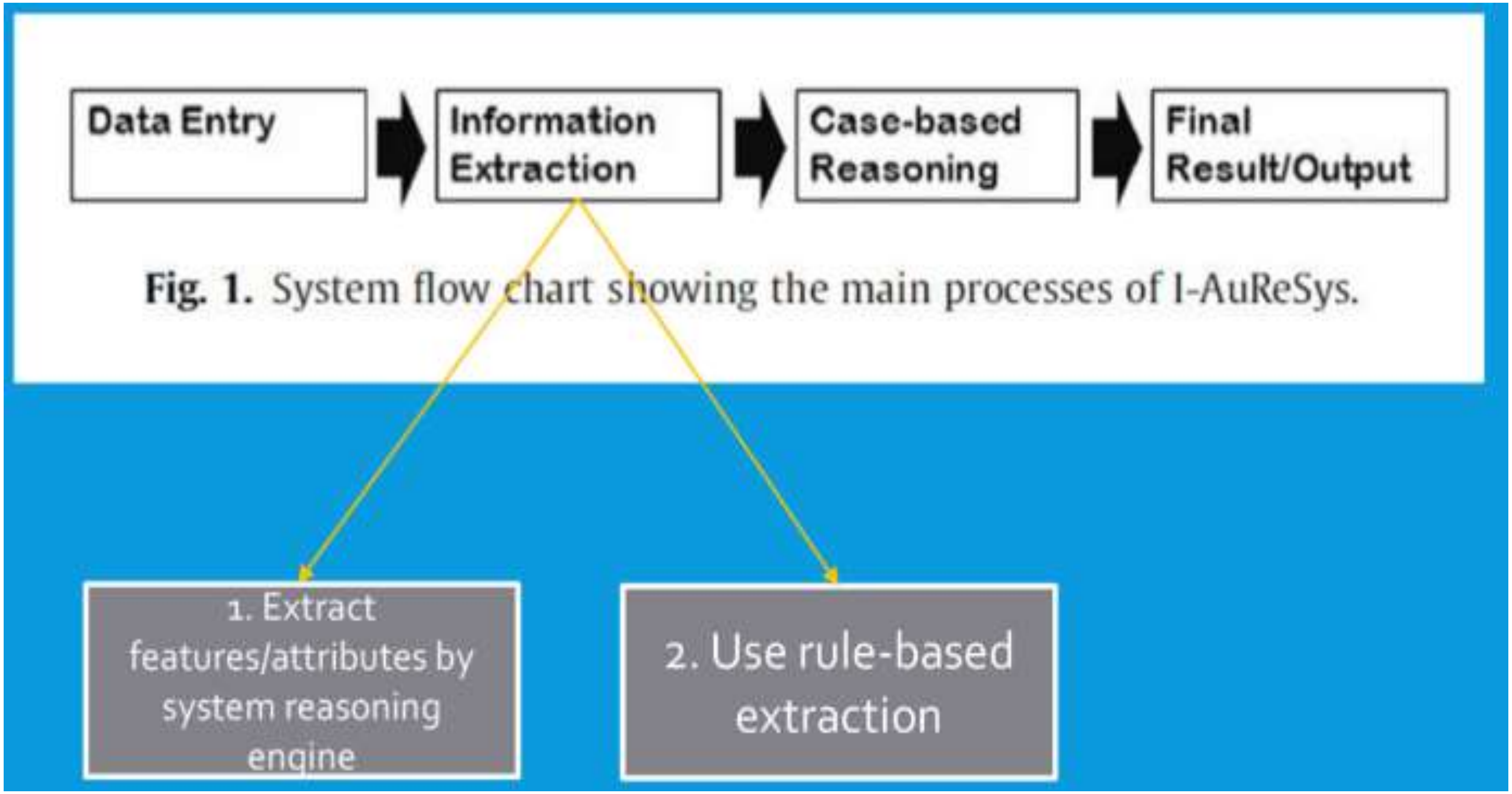
Fig. 1. System flow chart showing the main processes of I-AuReSys.

CONTD..

Step-2:


Data Entry
Information Extraction
Case-based Reasoning
Final Result/Output
Fig. 1. System flow chart showing the main processes of I-AuReSys.
1. Given in free text
2. Must adhere specific guidelines/format

Step-3:


Data Entry
Information Extraction
Case-based Reasoning
Final Result/Output
Fig. 1. System flow chart showing the main processes of I-AuReSys.
1. Extract features/attributes by system reasoning engine
2. Use rule-based extraction

Step-4:

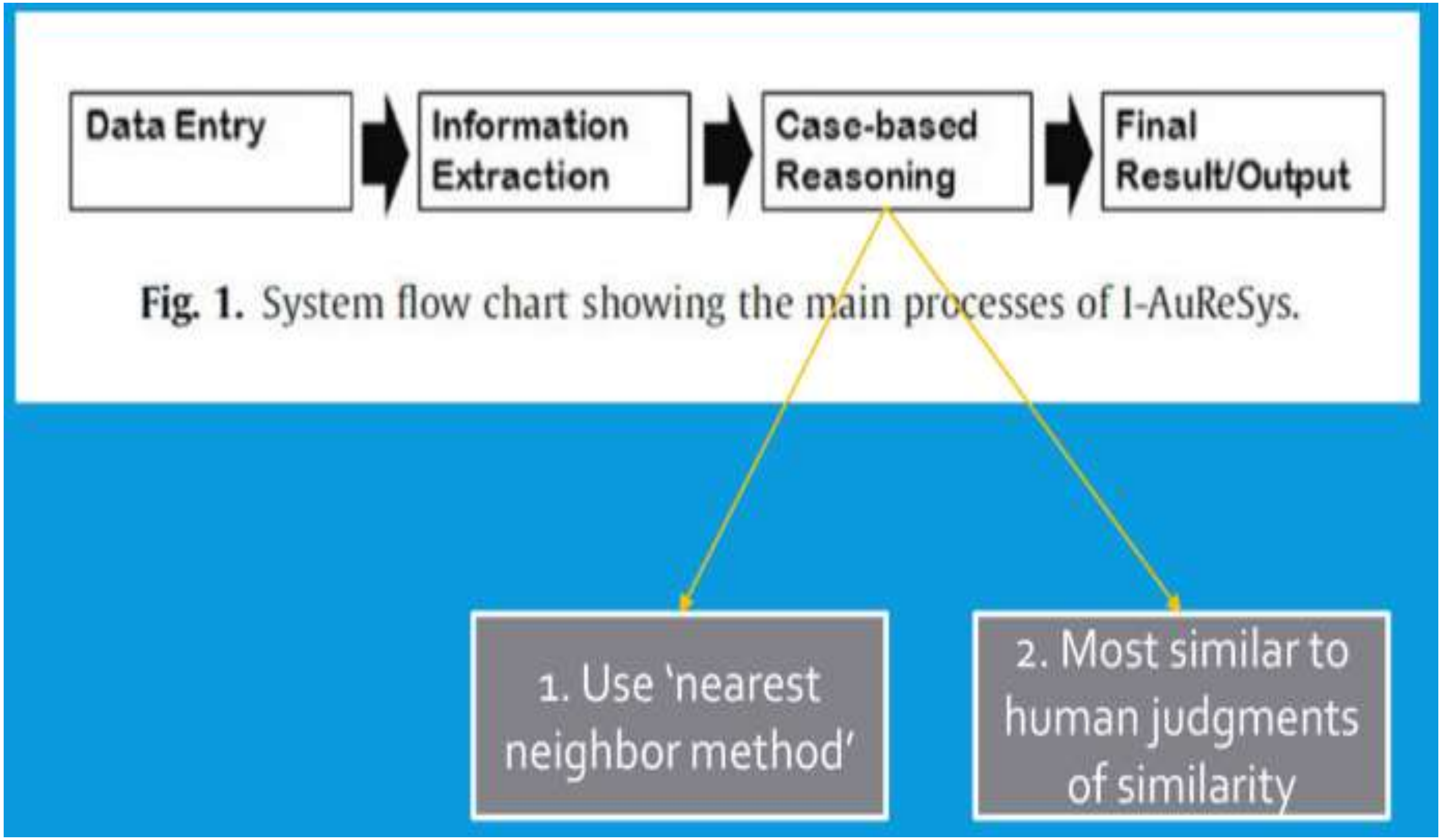
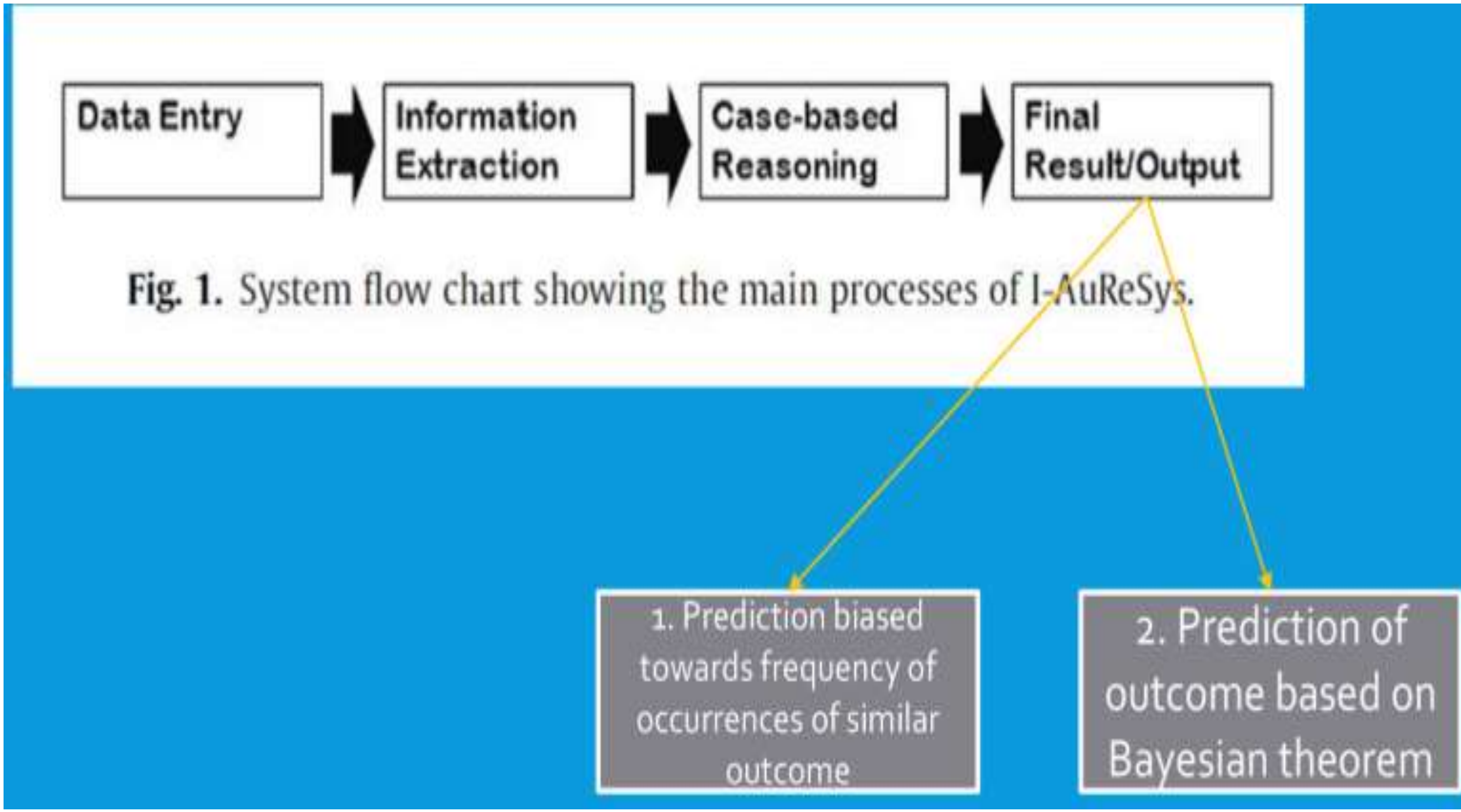


Fig. 1. System flow chart showing the main processes of I-AuReSys.

Step-5:


Data Entry
Information Extraction
Case-based Reasoning
Final Result/Output
Fig. 1. System flow chart showing the main processes of I-AuReSys.
1. Prediction biased towards frequency of occurrences of similar outcome
2. Prediction of outcome based on Bayesian theorem

INTELLIGENT AUTOPSY REPORT SYSTEM (I-AURESYS)

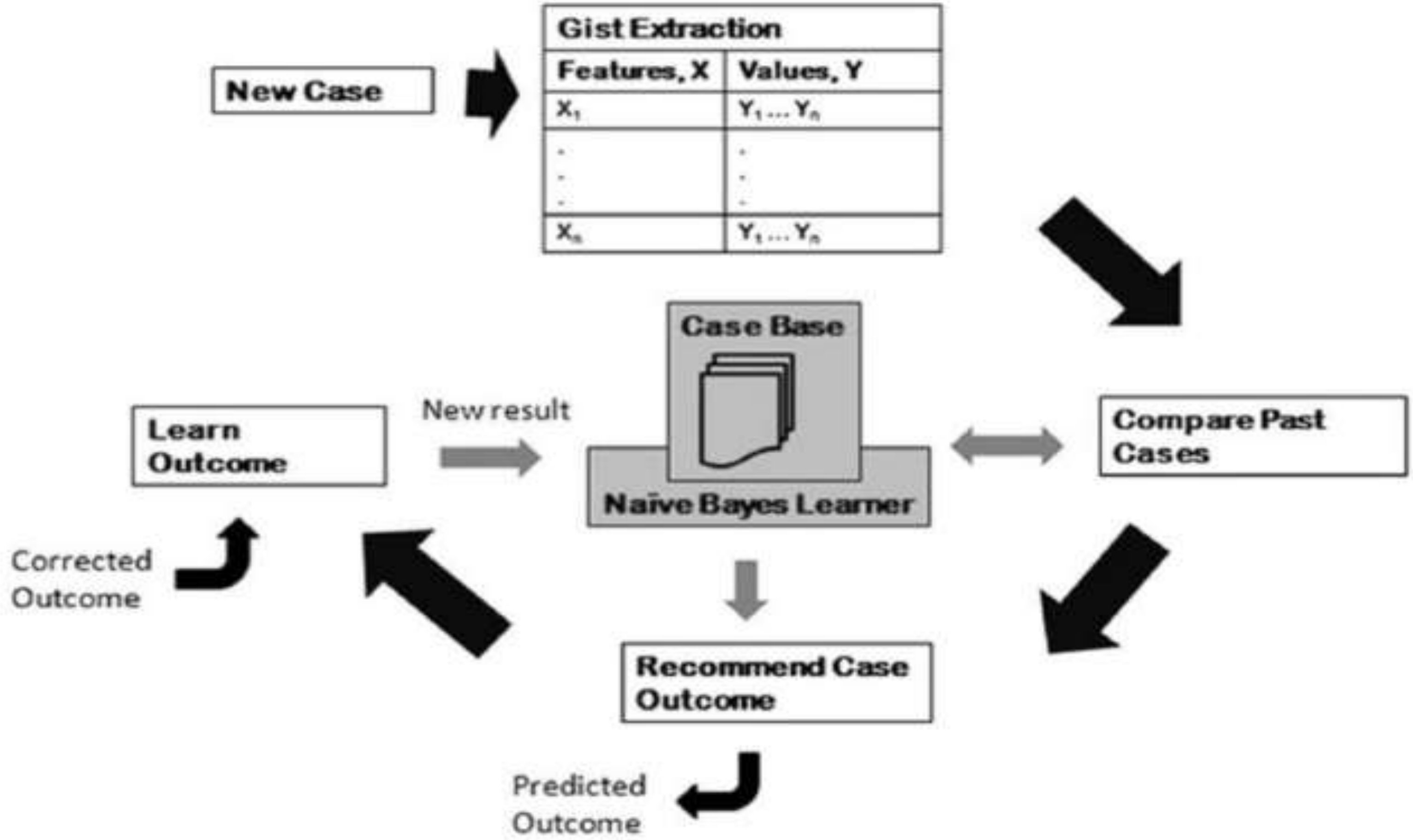


Fig. 3. CBR cycle of I-AuReSys.

Applications of CBR

- **Financial Decision Making:** CBR systems can be used in **financial institutions** to help make **decisions on loan approvals, risk assessments, and investment strategies** by **comparing past cases with current situations.**
- **Legal Reasoning:** Case-Based Reasoning in machine learning systems can be used in the l**egal field to assist with case law research** and the **preparation of legal arguments** by **retrieving and adapting cases with similar legal issues.**
- **Transportation**: CBR systems can be used in **transportation** to **optimize routing, scheduling, and resource allocation** by **learning from past cases.**

THANK YOU