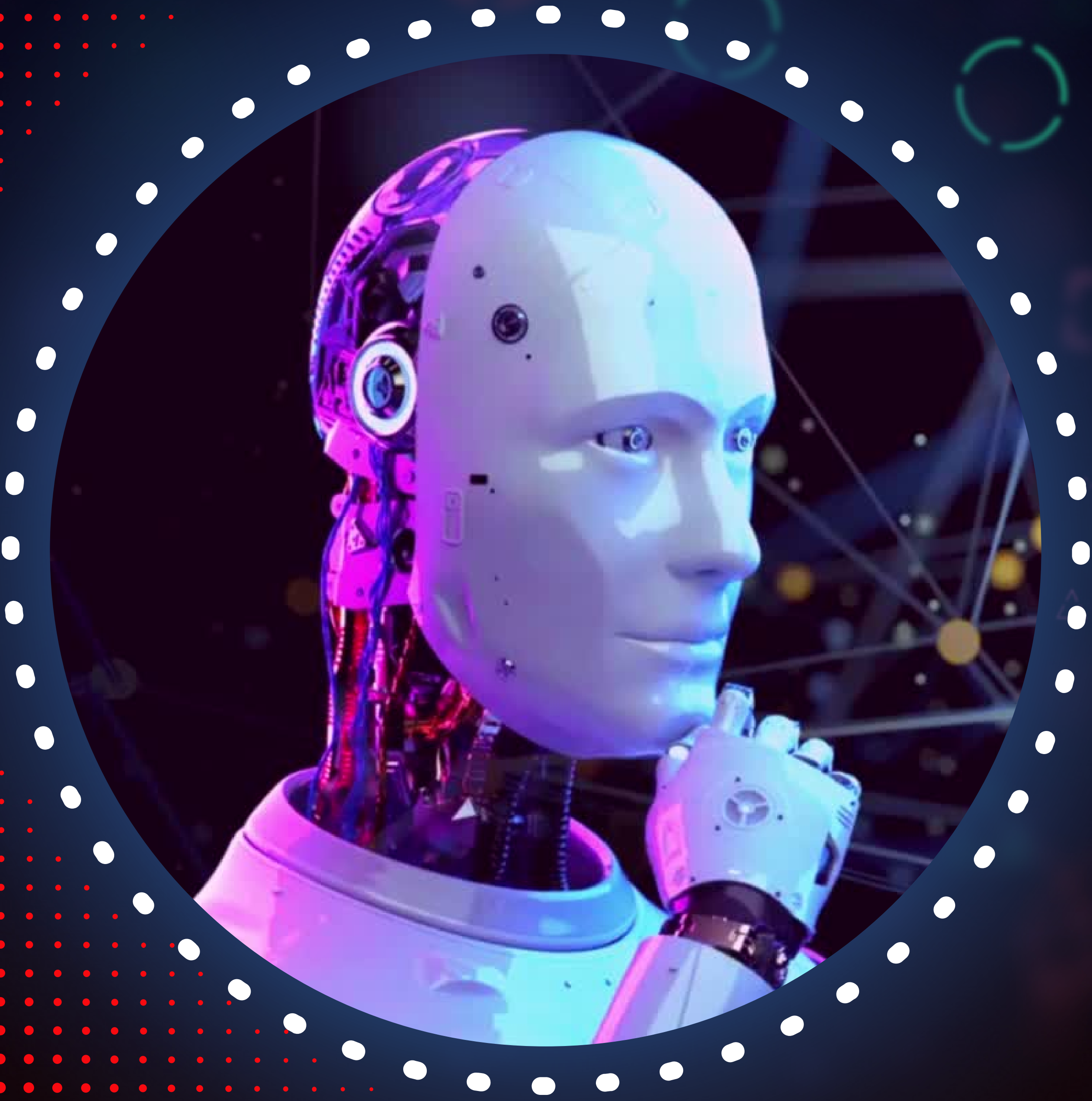
Pantech e Learning
DIGITAL LEARNING SIMPLIFIED



MACHINE LEARNING

DAY – 15

SUPPORT VECTOR MACHINE

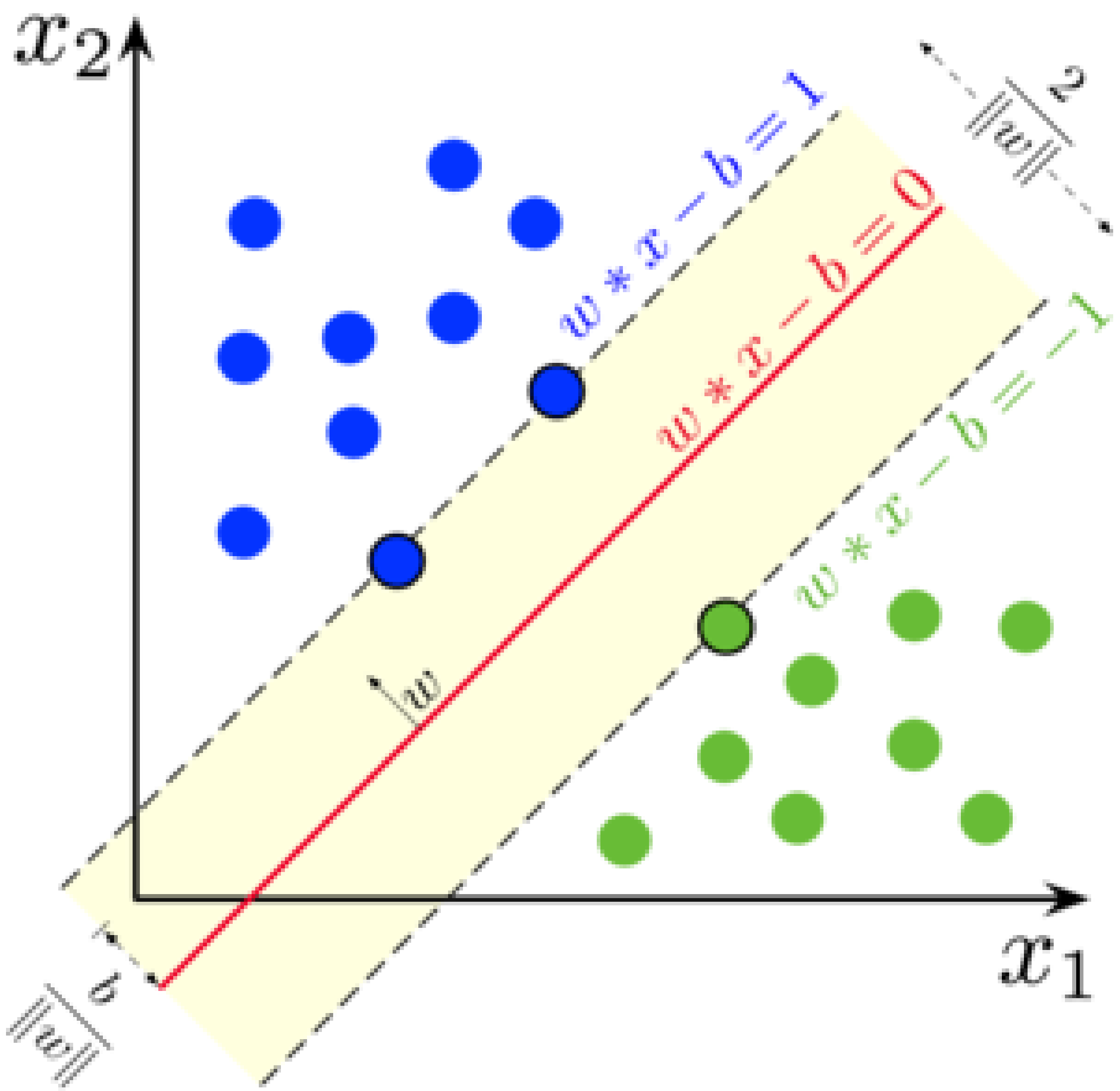
SUPPORT VECTOR MACHINE

- Support Vector Machines (SVMs) are a popular and powerful supervised learning algorithm used for classification and regression analysis.

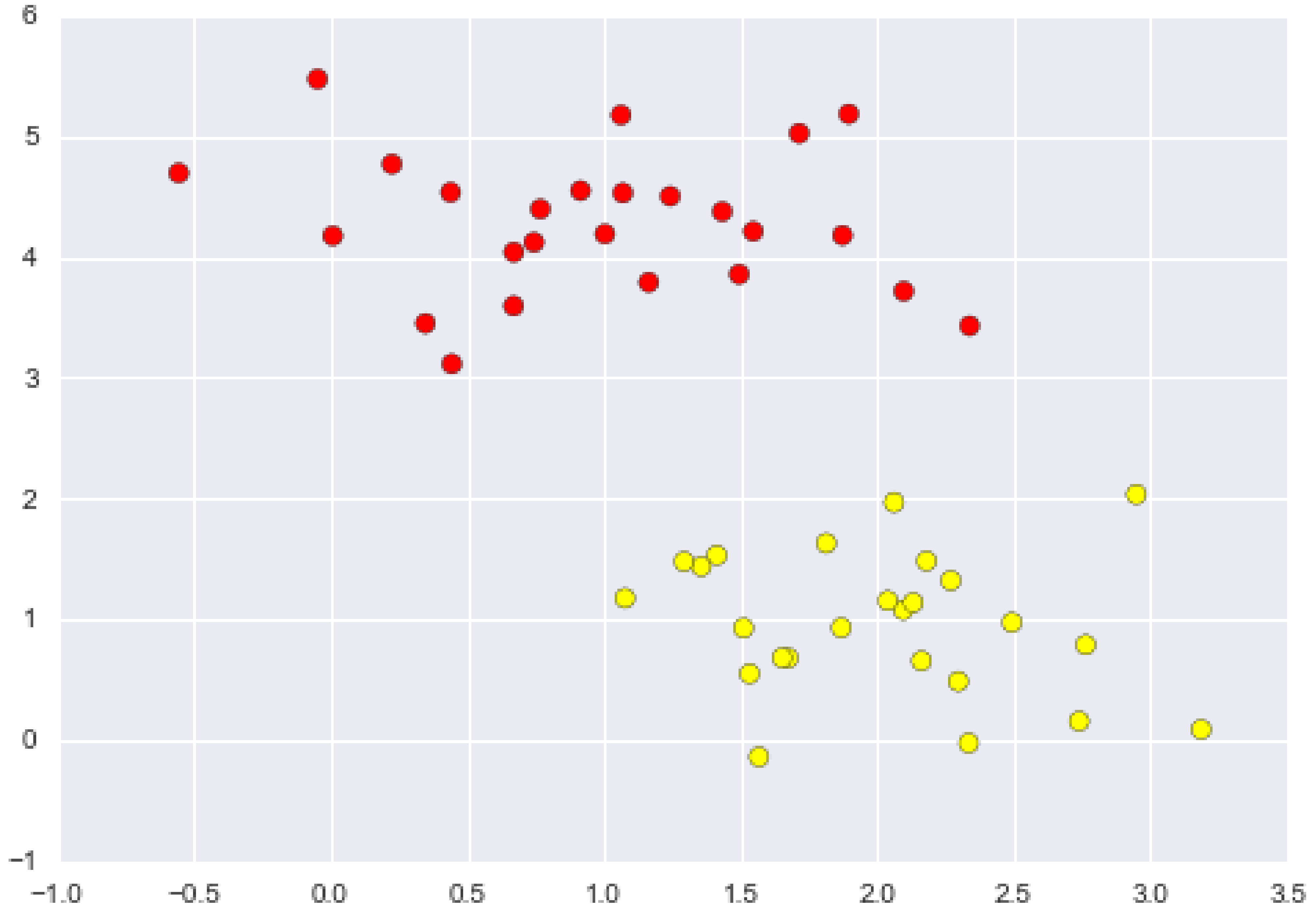
- SVMs are based on the idea of finding the hyperplane that best separates the data into different classes, by maximizing the margin between the hyperplane and the closest data points.

- SVM is a very versatile algorithm and can handle both linear and nonlinear datasets.

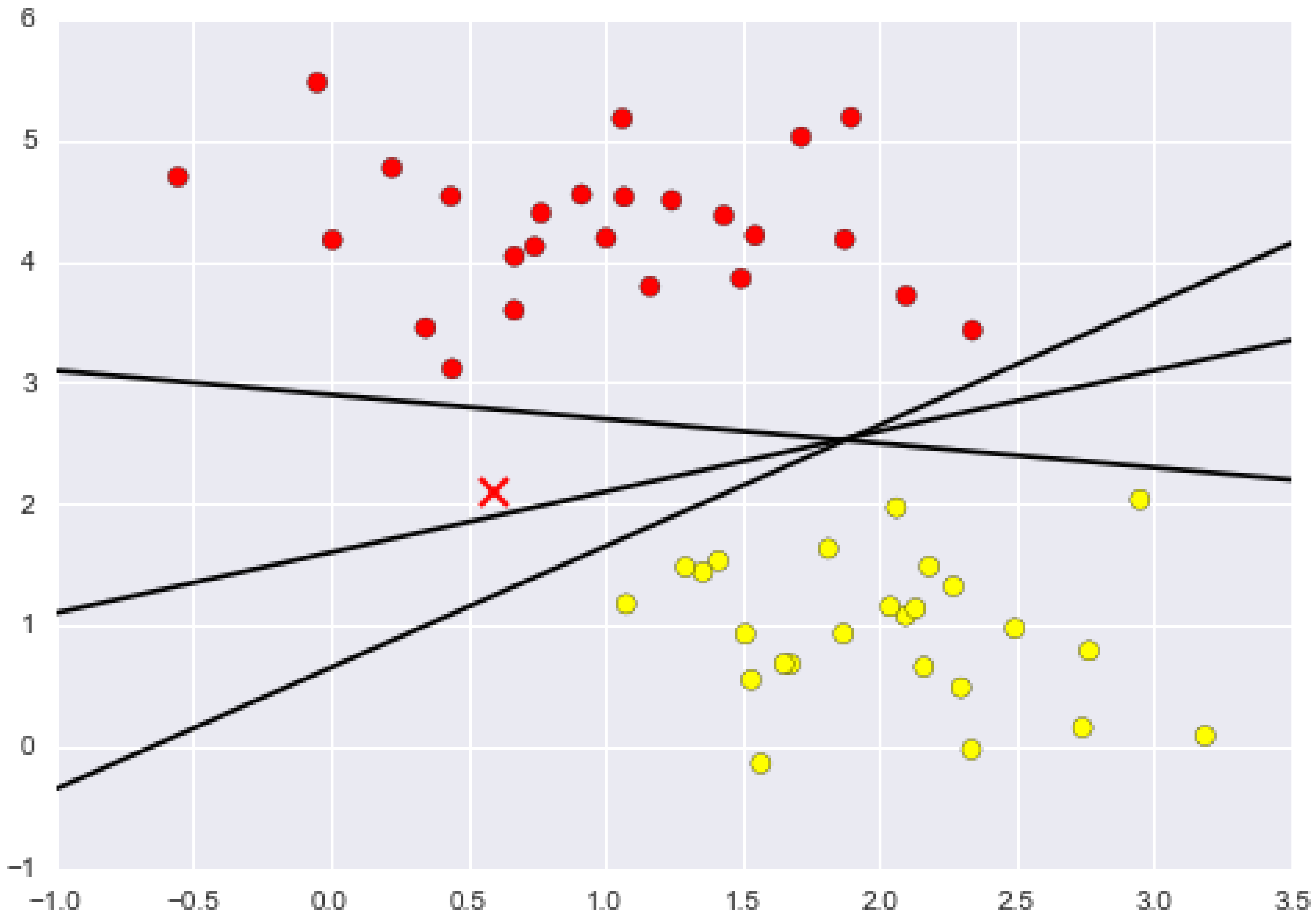
SUPPORT VECTOR MACHINE - CLASSIFIER



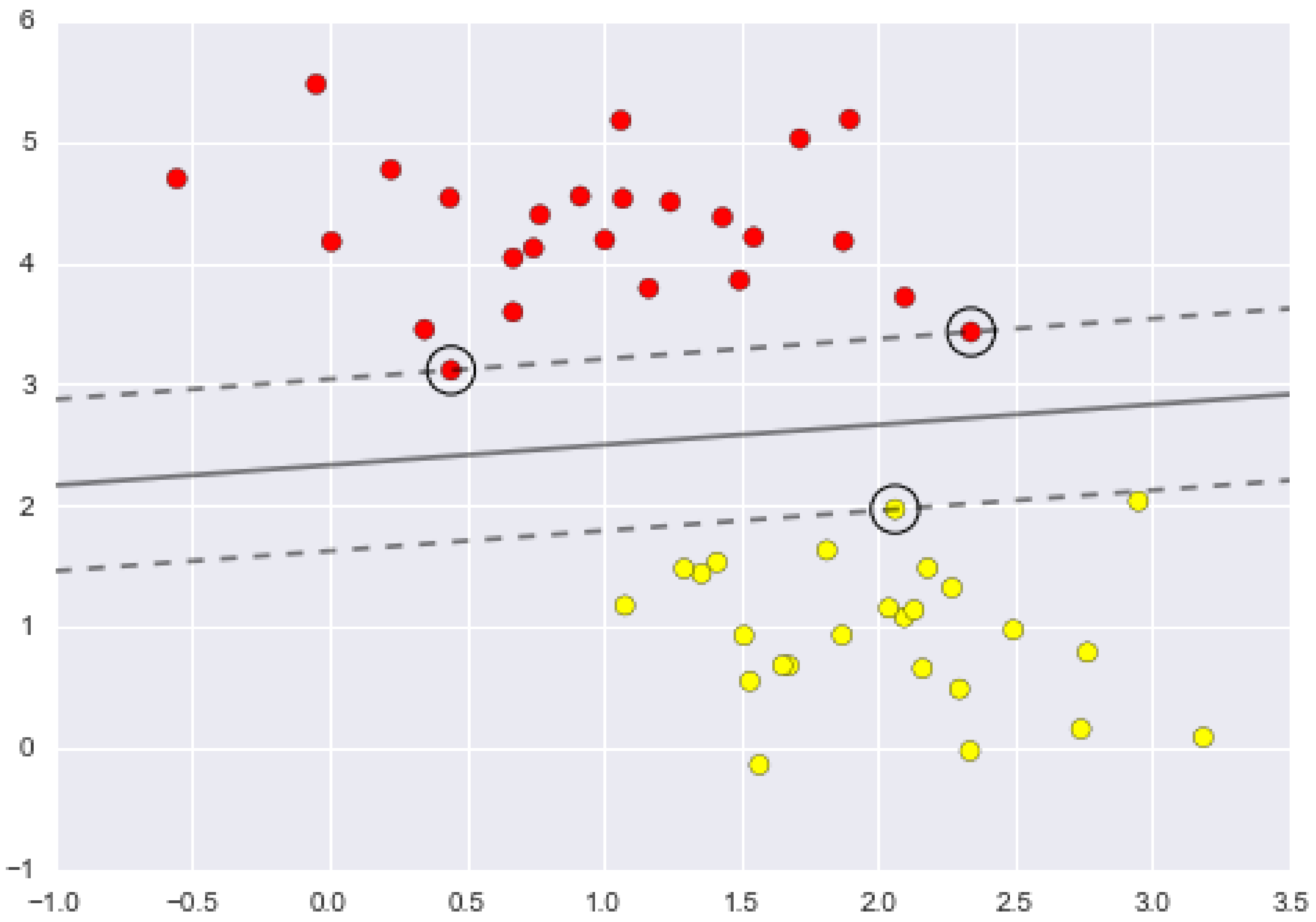
SUPPORT VECTOR MACHINE - CLASSIFIER



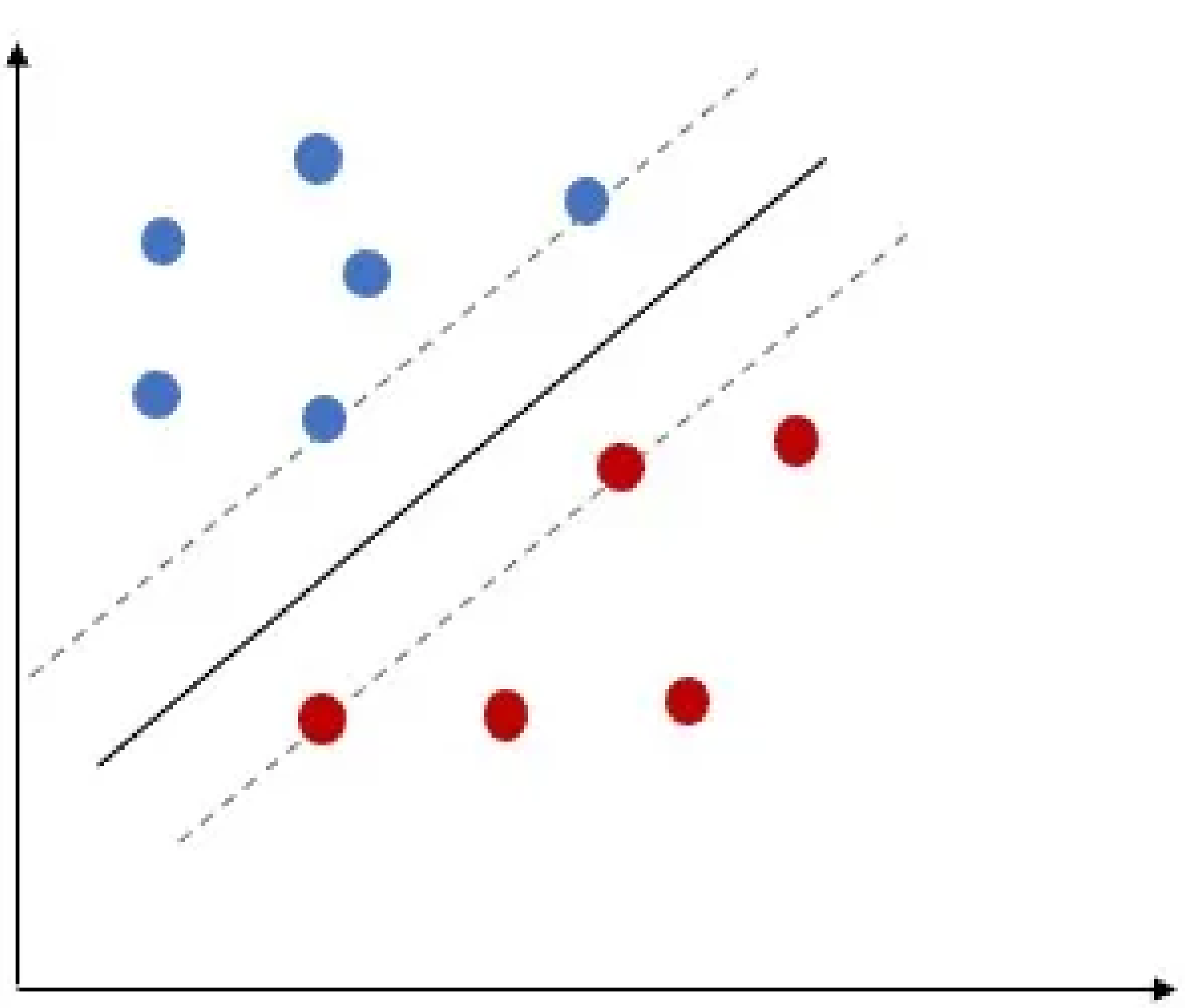
SUPPORT VECTOR MACHINE - CLASSIFIER

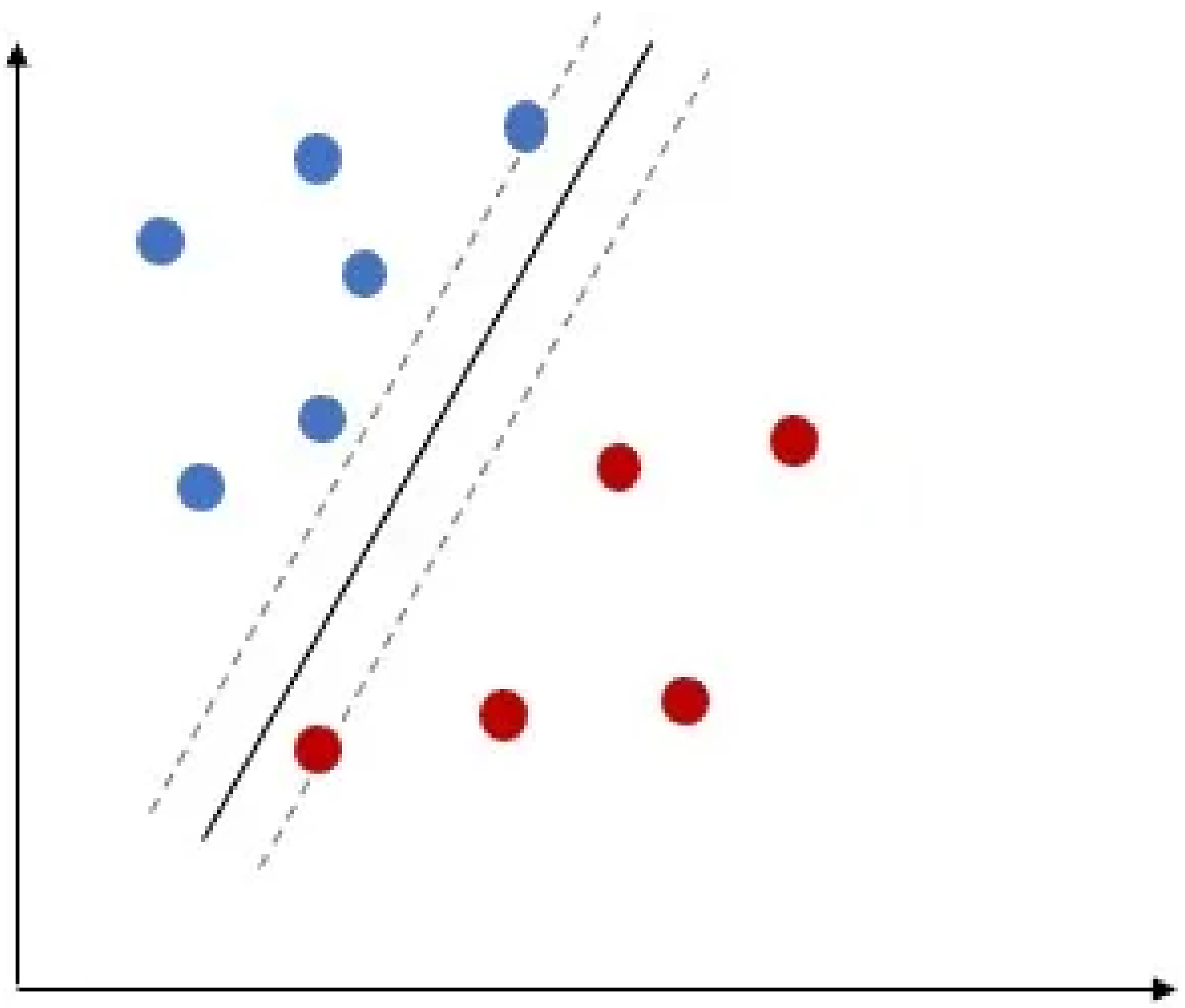


SUPPORT VECTOR MACHINE - CLASSIFIER



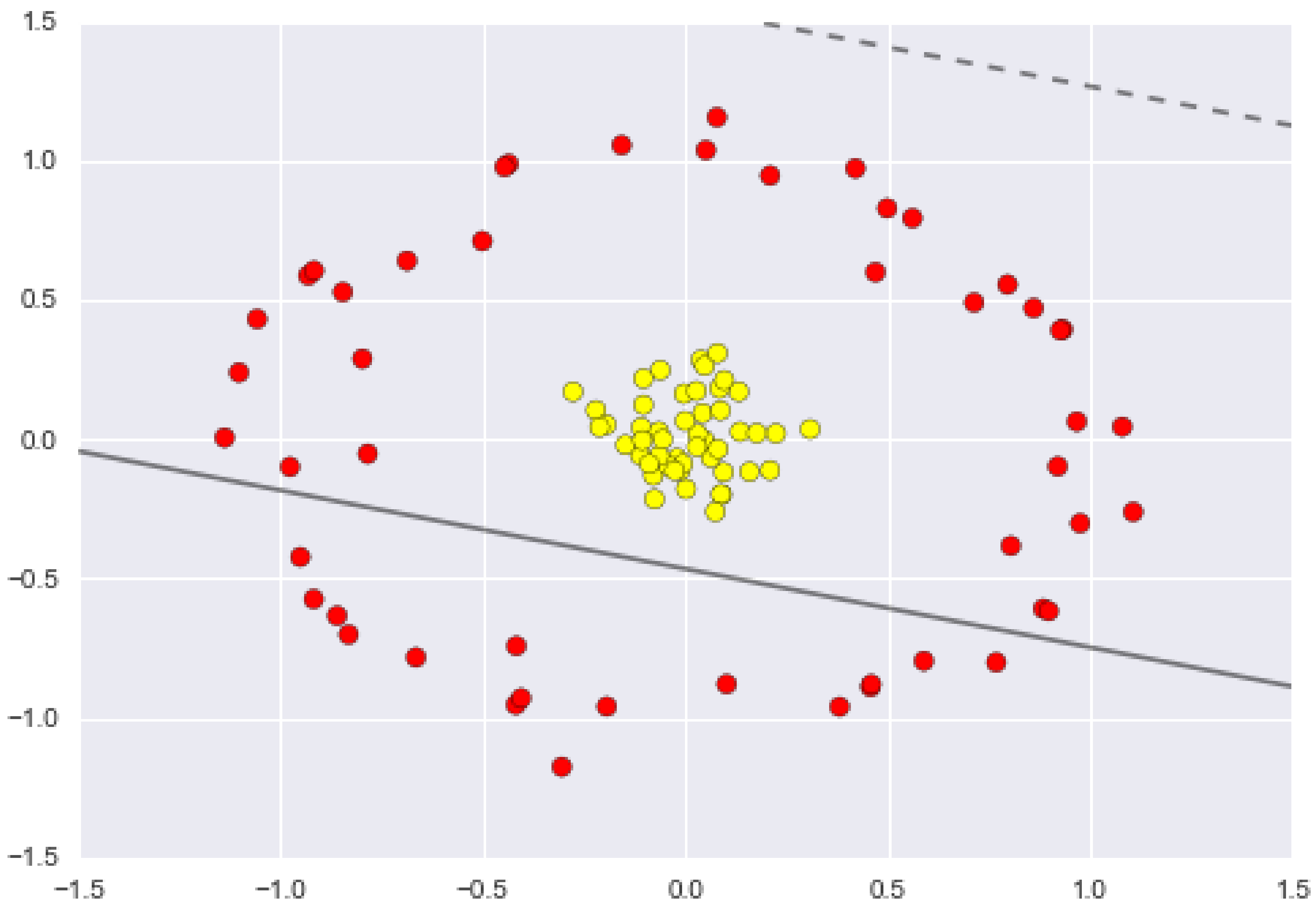
SUPPORT VECTOR MACHINE - CLASSIFIER



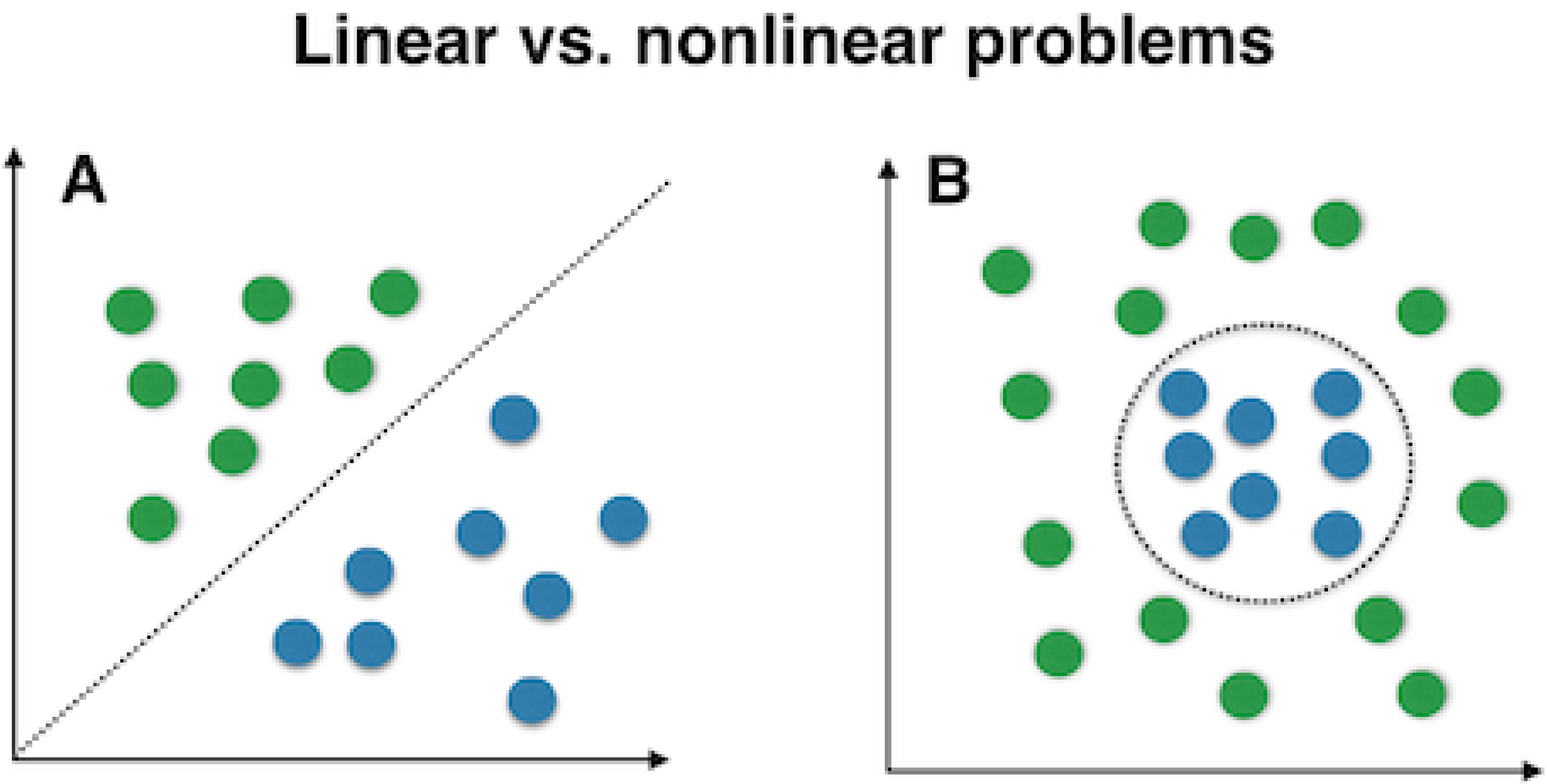


MAXIMUM MARGIN IS SELECTED

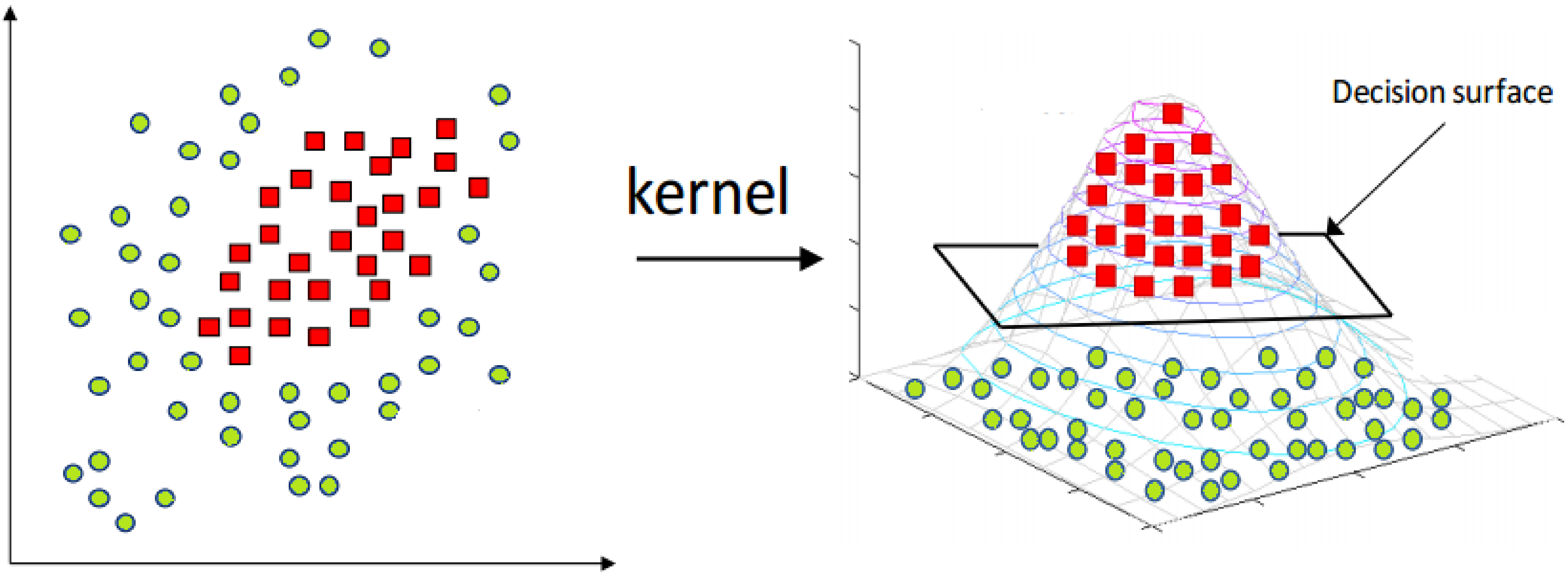
SUPPORT VECTOR MACHINE - CLASSIFIER



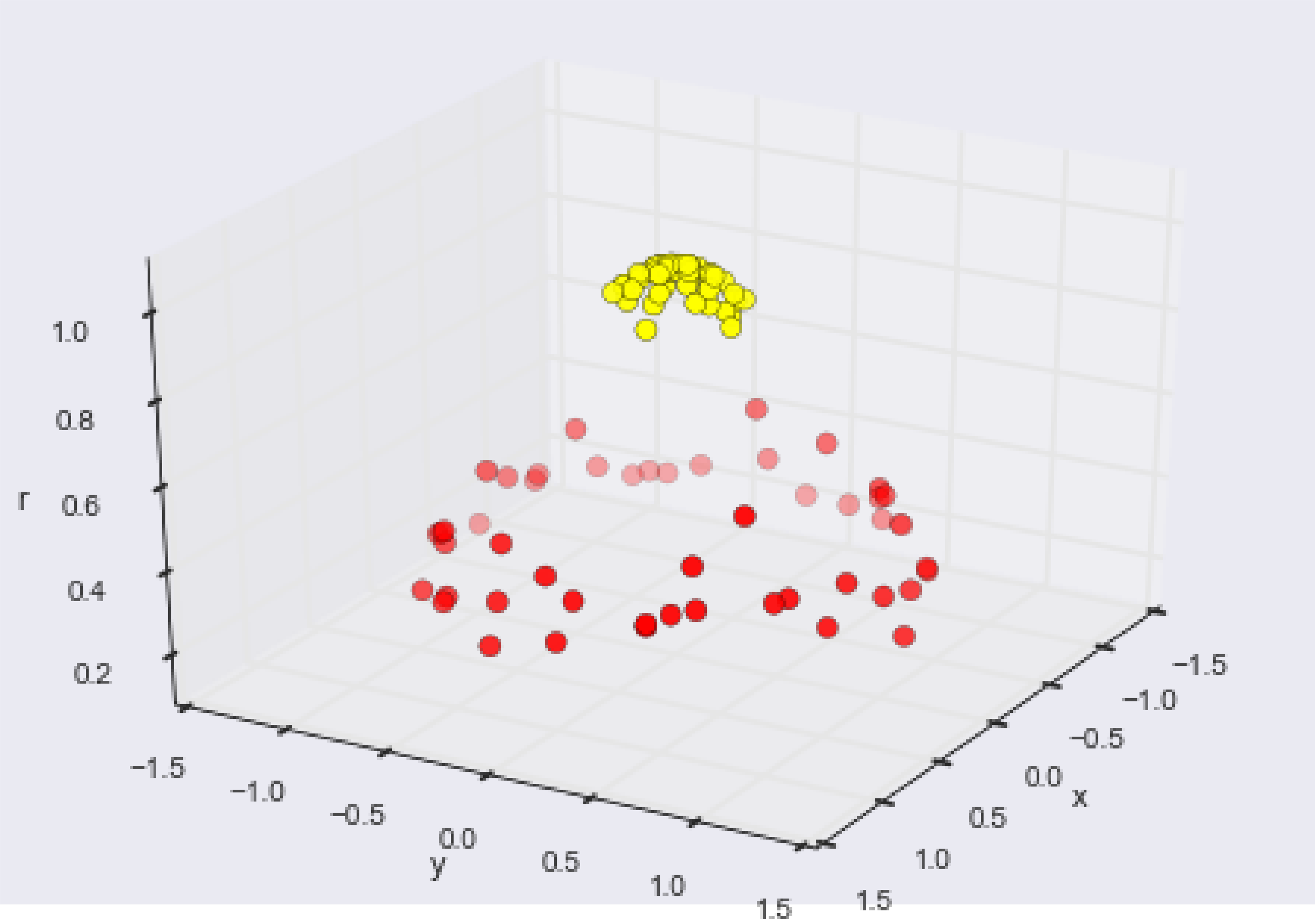
LINEAR VS NON-LINEAR



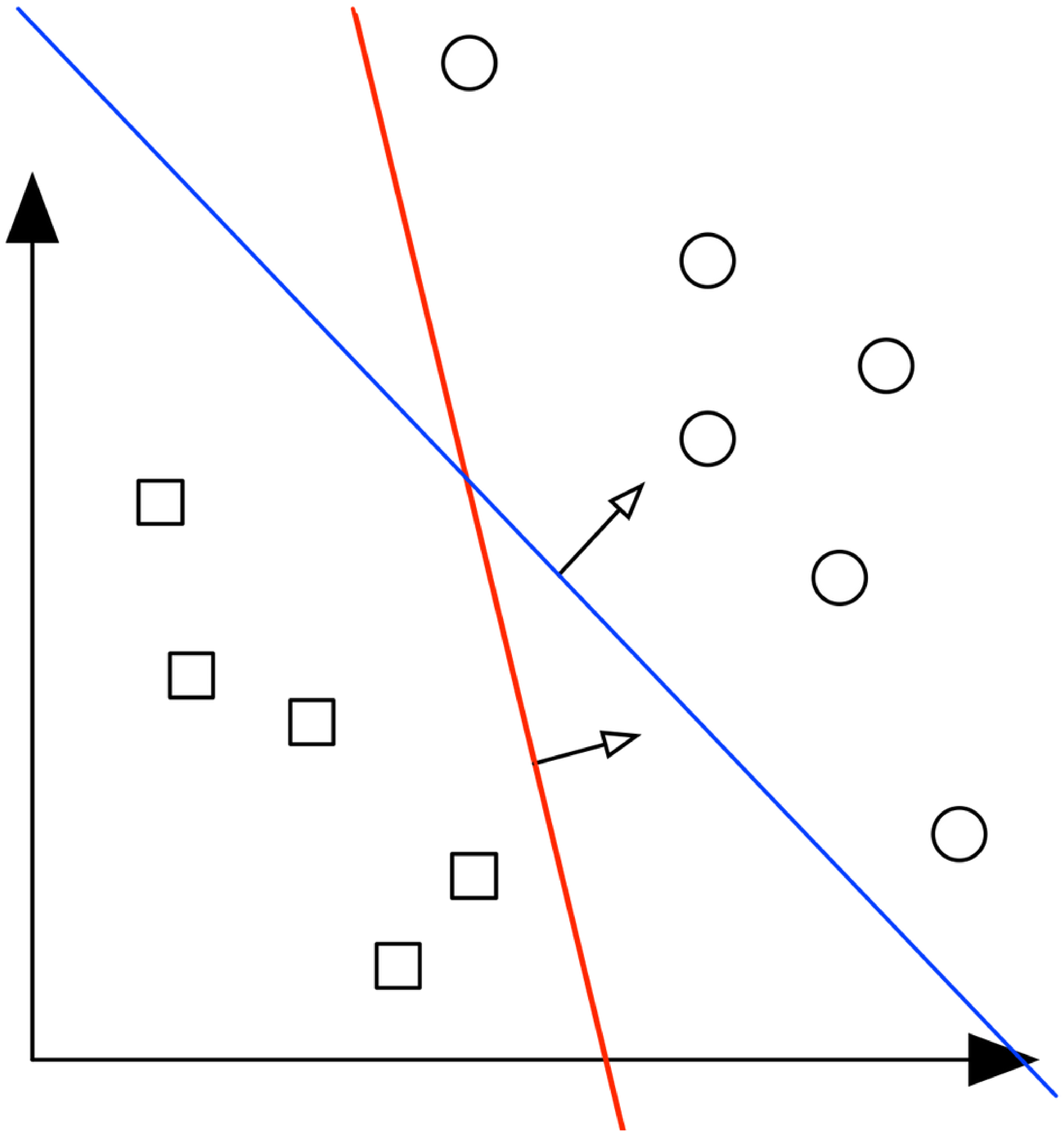
KERNEL IN SVM

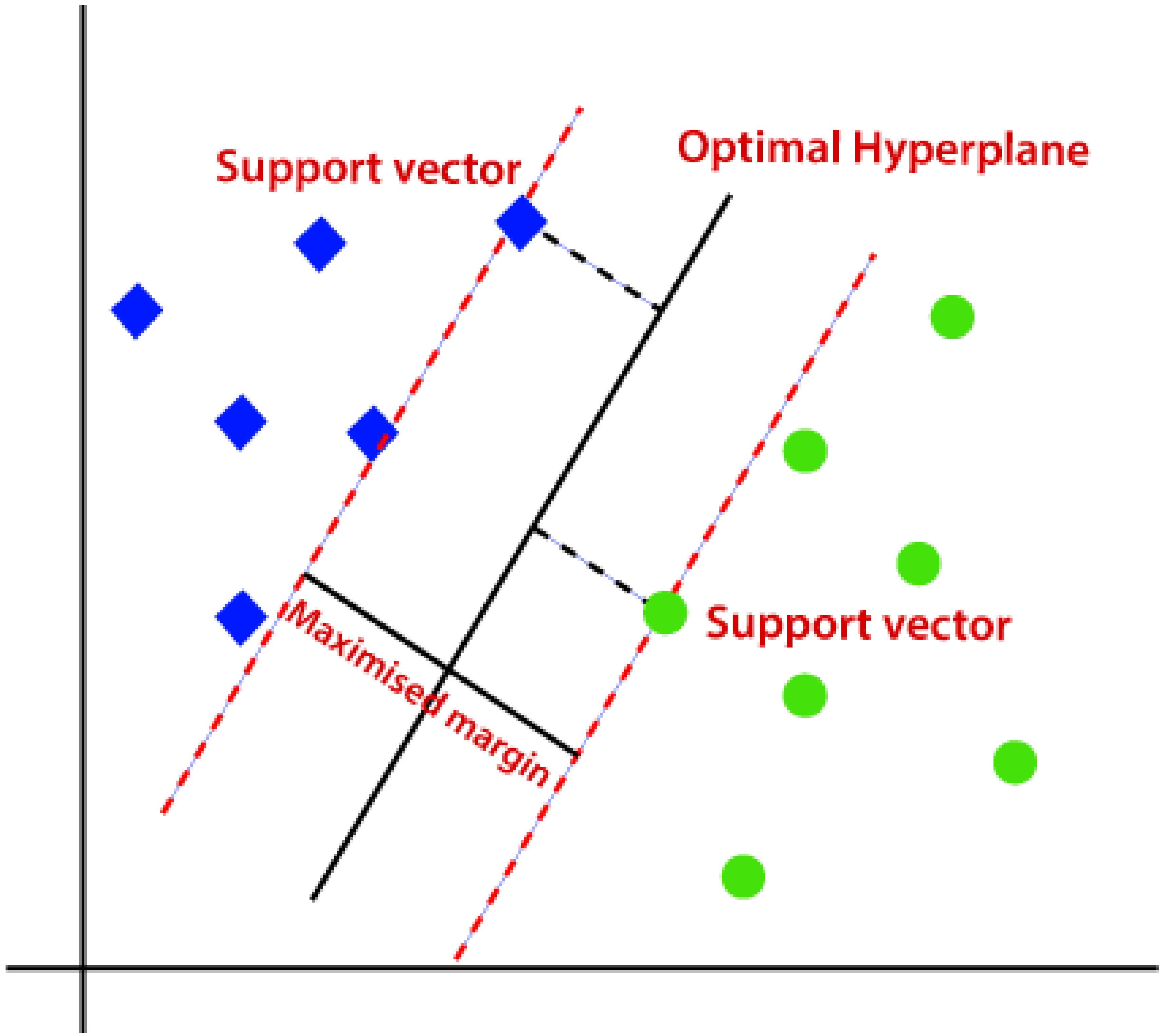


KERNEL IN SVM

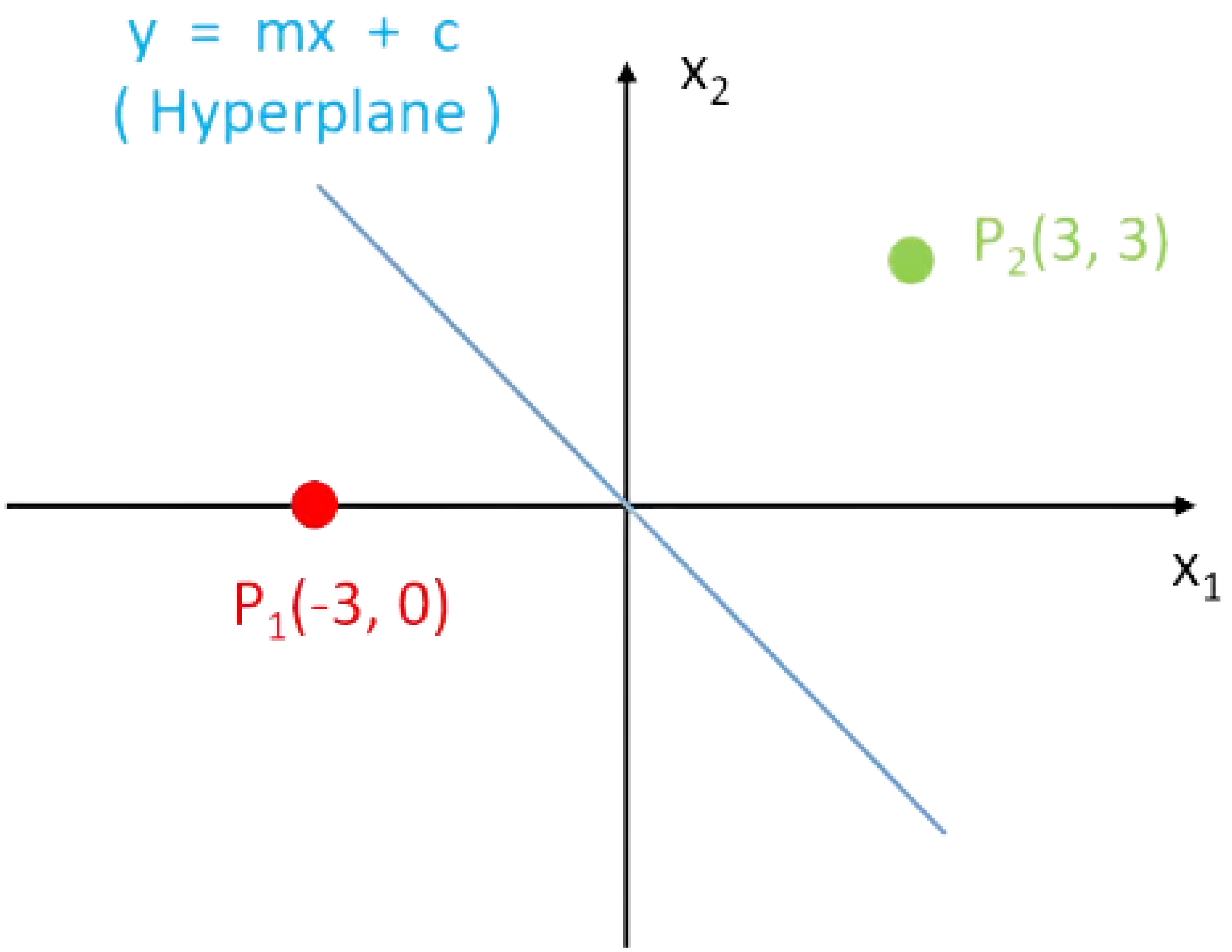


HOW SVM WORKS?





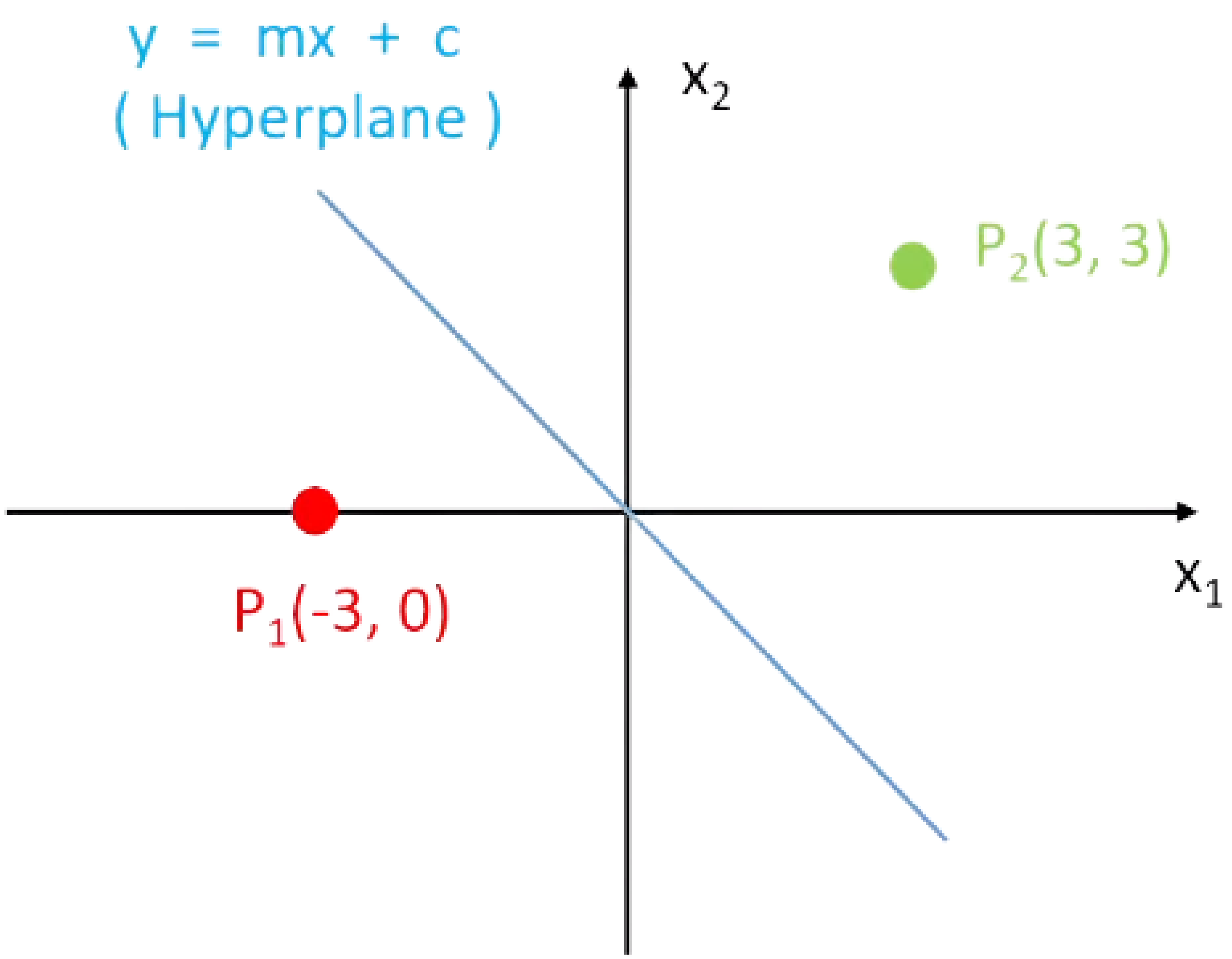
HOW SVM WORKS?



Let slope, m = - 1

Intercept, c = 0

HOW SVM WORKS?



Let slope, m = - 1

Intercept, c = 0

w --> parameters of the line

(m, c) = (-1, 0)

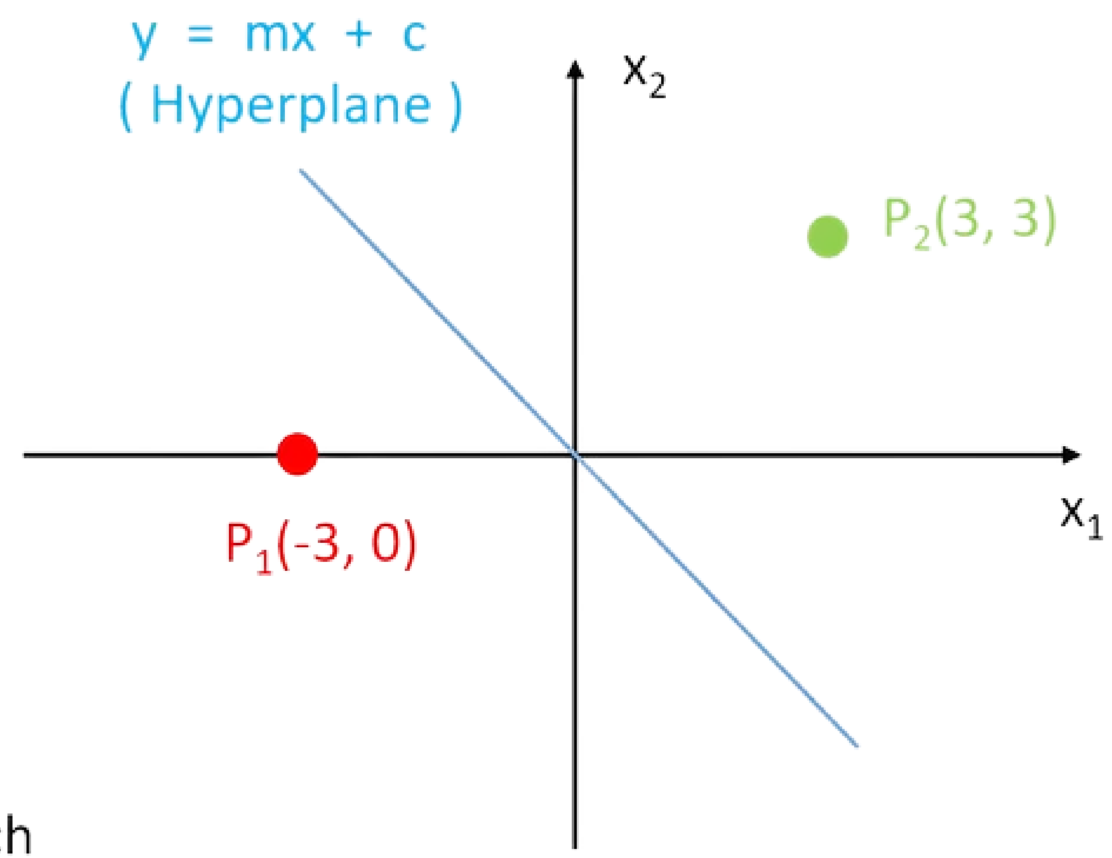
HOW SVM WORKS?

$P_1(-3, 0)$

$$w^T x = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} -3 & 0 \end{bmatrix}$$

$w^T x = 3$

(Positive)



$P_2(3, 3)$

$$w^T x = \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} 3 & 3 \end{bmatrix}$$

$w^T x = -3$

(Negative)

Inference: For all the points which lie in the left side of the hyperplane, $w^T x$ value will be Positive

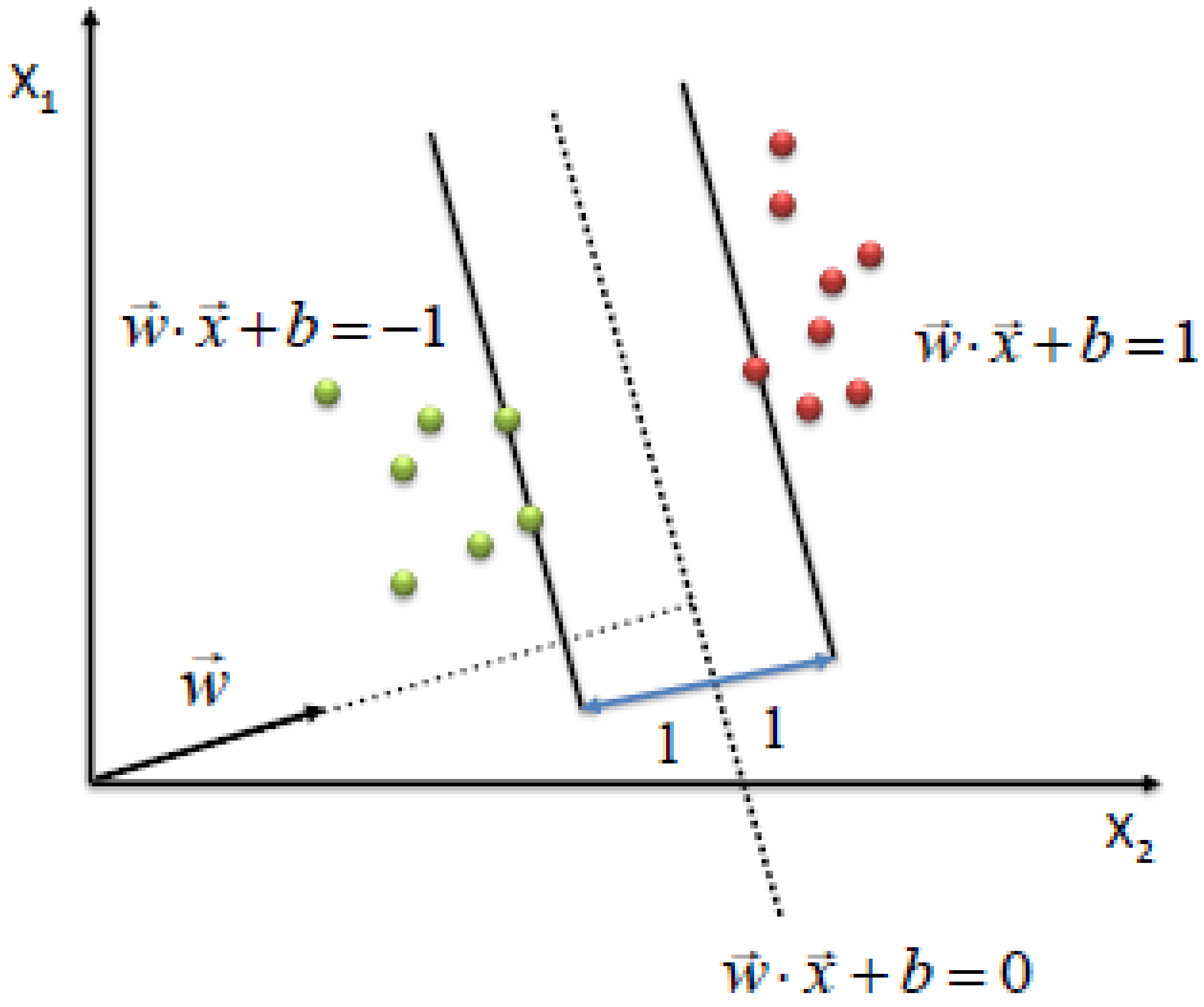
Let slope, m = - 1

Intercept, c = 0

w --> parameters of the line
(m, c) = (-1, 0)

Inference: For all the points which lie in the right side of the hyperplane, $w^T x$ value will be Negative

CLASSIFICATION



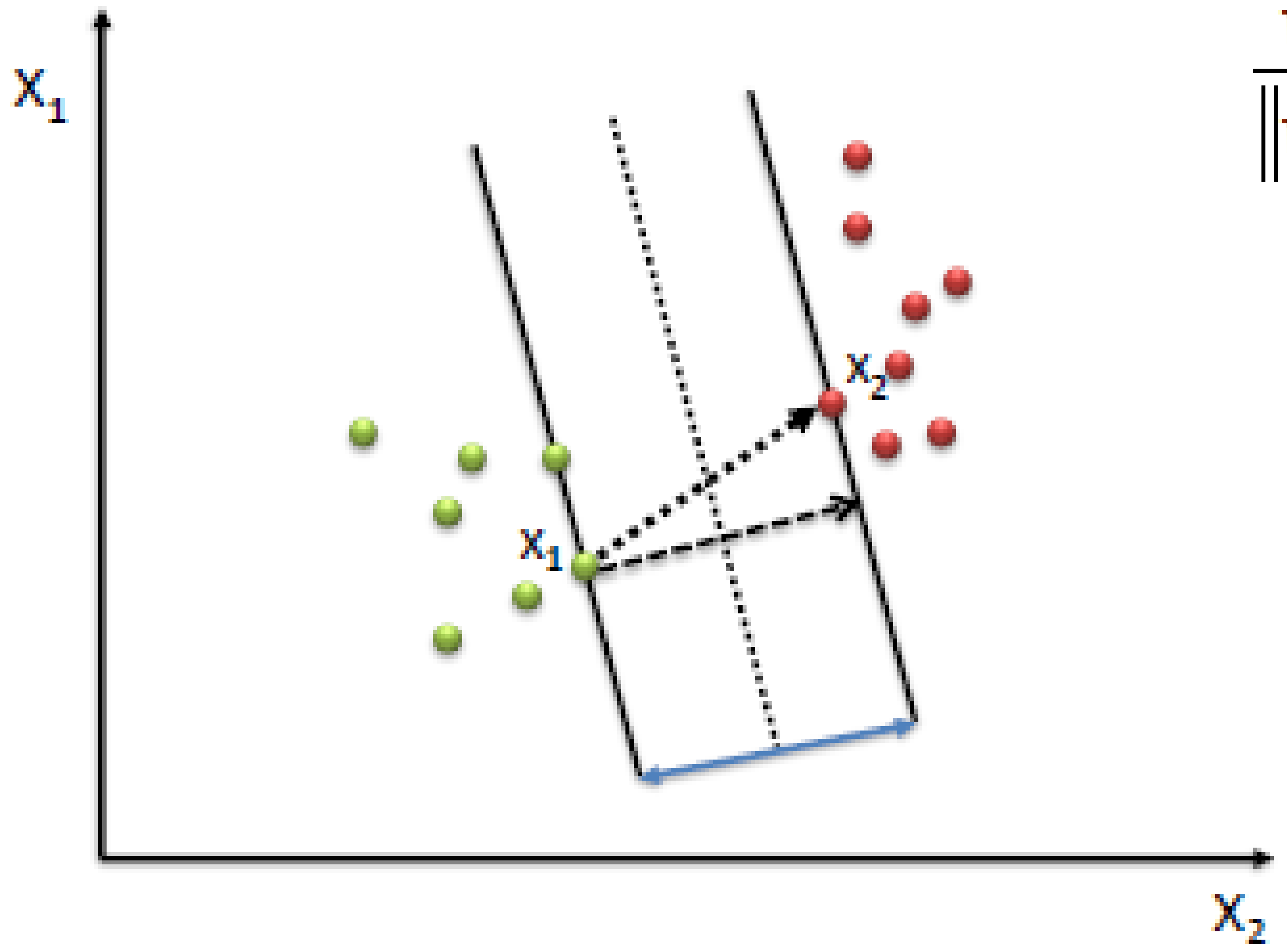
$$\boxed{\max \frac{2}{\|w\|}}$$

s.t.

$(w\cdot x+b)\geq 1, \forall x \text{ of class } 1$

$(w\cdot x+b)\leq -1, \forall x \text{ of class } 2$

MAXIMUM MARGIN CALCULATION



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

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

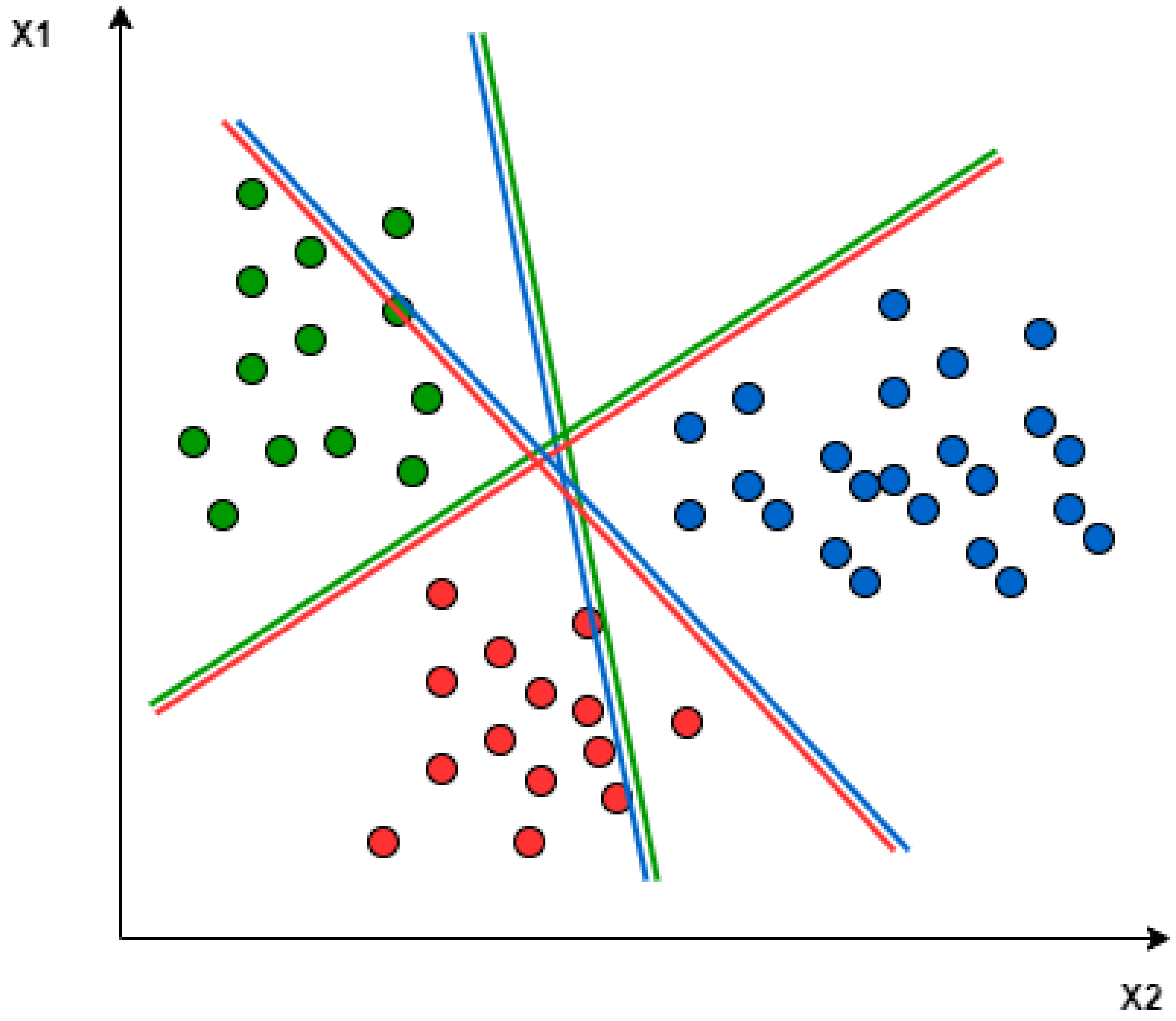
$$w\cdot x_1 + b = -1$$

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

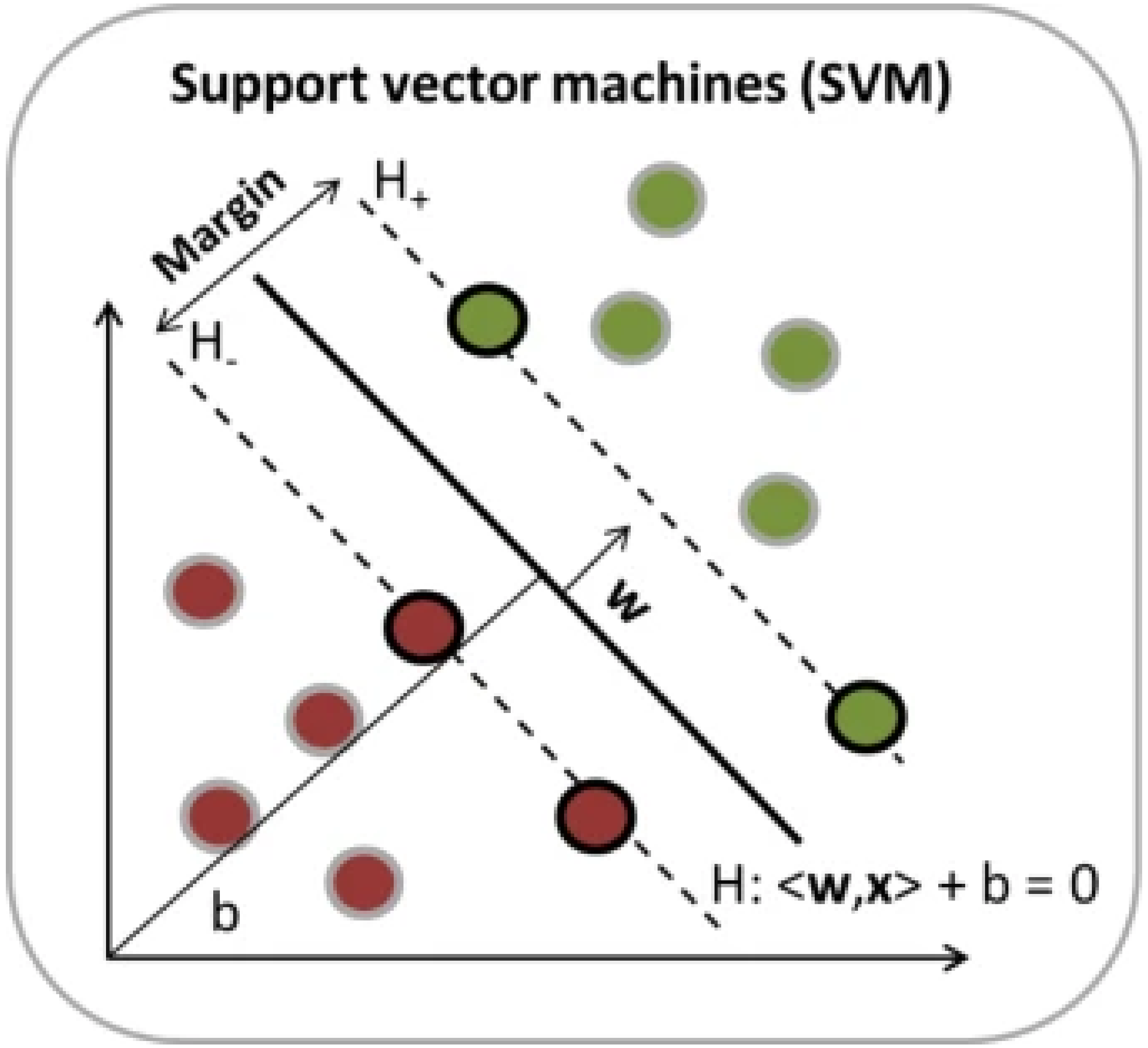
$$w\cdot x_2 - w\cdot x_1 = 2$$

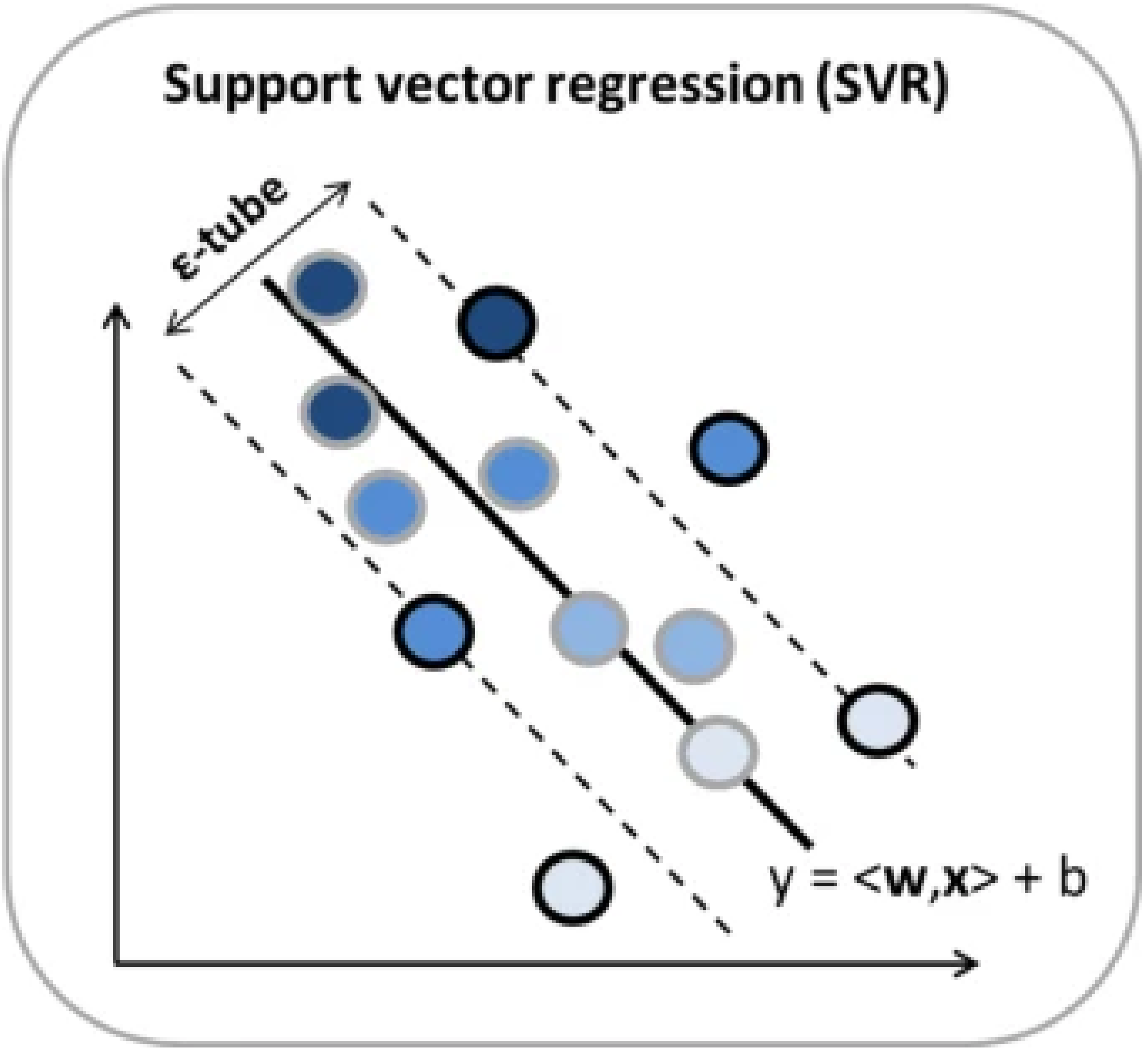
$$\frac{w}{\|w\|}(x_2 - x_1) = \frac{2}{\|w\|}$$

MULTICLASS CLASSIFICATION IN SVM

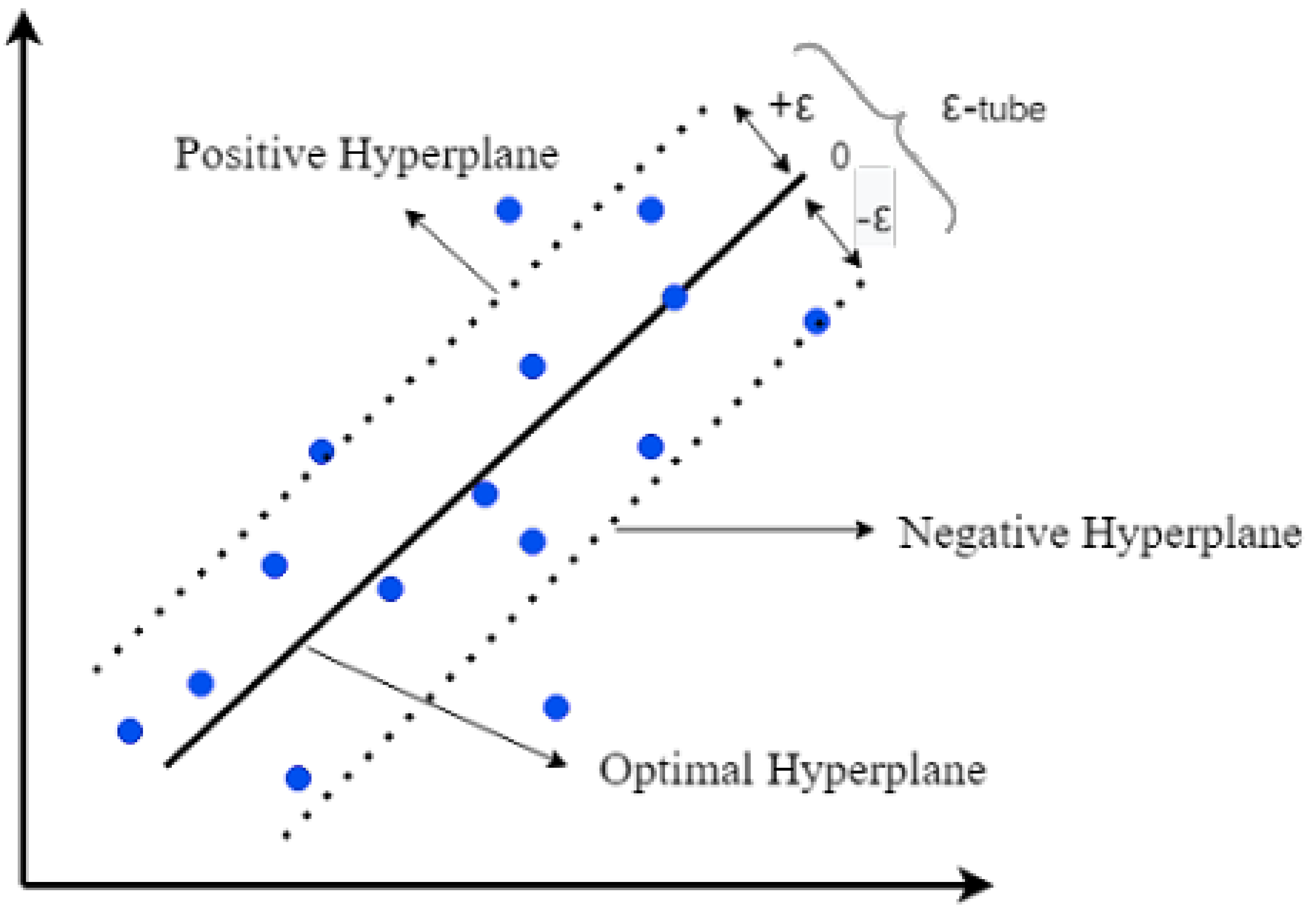


SUPPORT VECTOR REGRESSION





SUPPORT VECTOR REGRESSION



$$\min \frac{1}{2}||w||^2$$

SUPPORT VECTOR MACHINES

```
from sklearn.svm import SVC, SVR

classifier_model  = SVC()

regressor_model = SVR()
```

SUPPORT VECTOR MACHINES KERNEL PARAMETER

1.4.6. Kernel functions

The *kernel function* can be any of the following:

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma \langle x, x' \rangle + r)^d$, where d is specified by parameter `degree`, r by `coef0`.
- rbf: $\exp(-\gamma \|x - x'\|^2)$, where γ is specified by parameter `gamma`, must be greater than 0.
- sigmoid $\tanh(\gamma \langle x, x' \rangle + r)$, where r is specified by `coef0`.

Different kernels are specified by the `kernel` parameter:

```
>>> linear_svc = svm.SVC(kernel='linear')
>>> linear_svc.kernel
'linear'
>>> rbf_svc = svm.SVC(kernel='rbf')
>>> rbf_svc.kernel
'rbf'
```

DECISION TREE

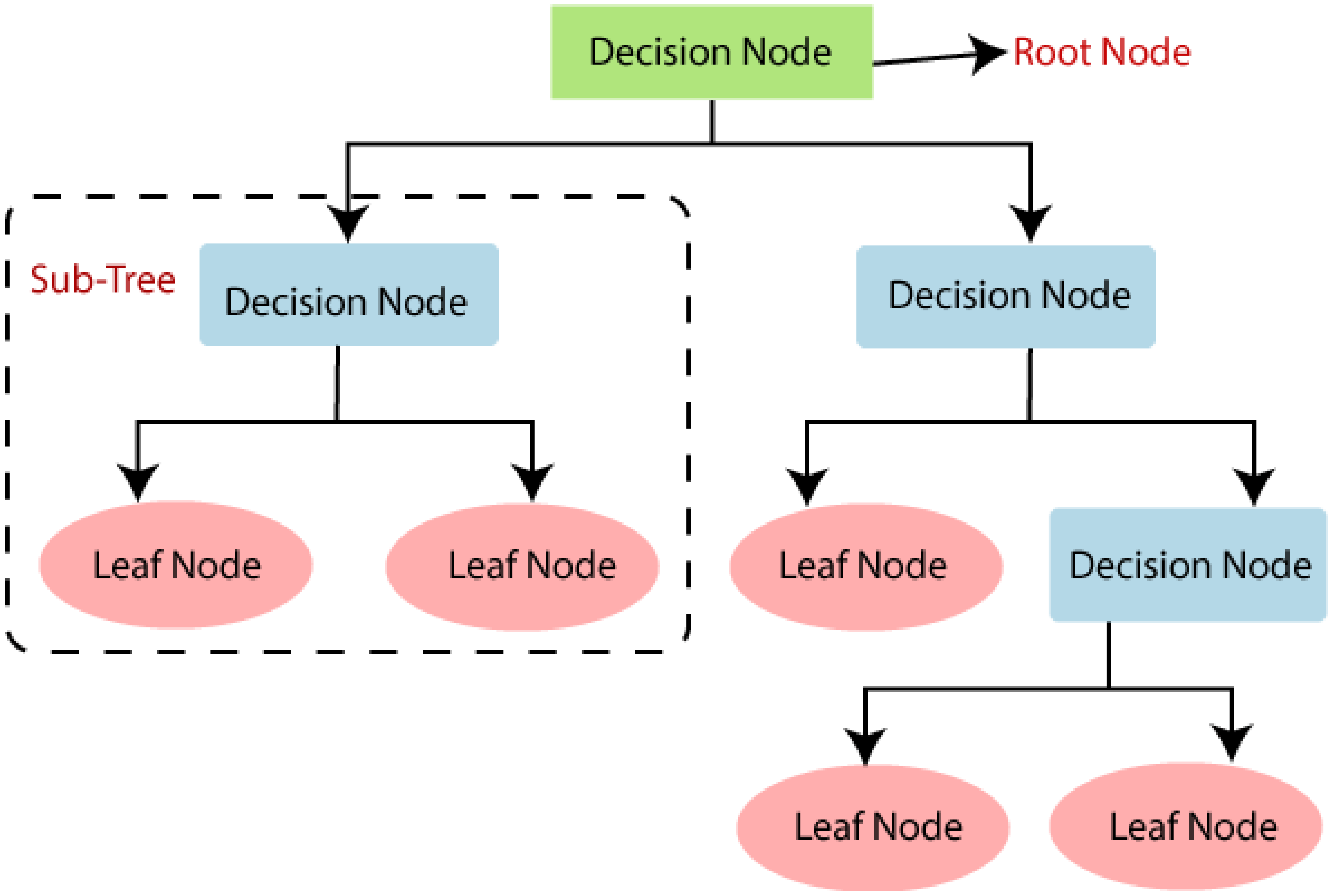
DECISION TREE

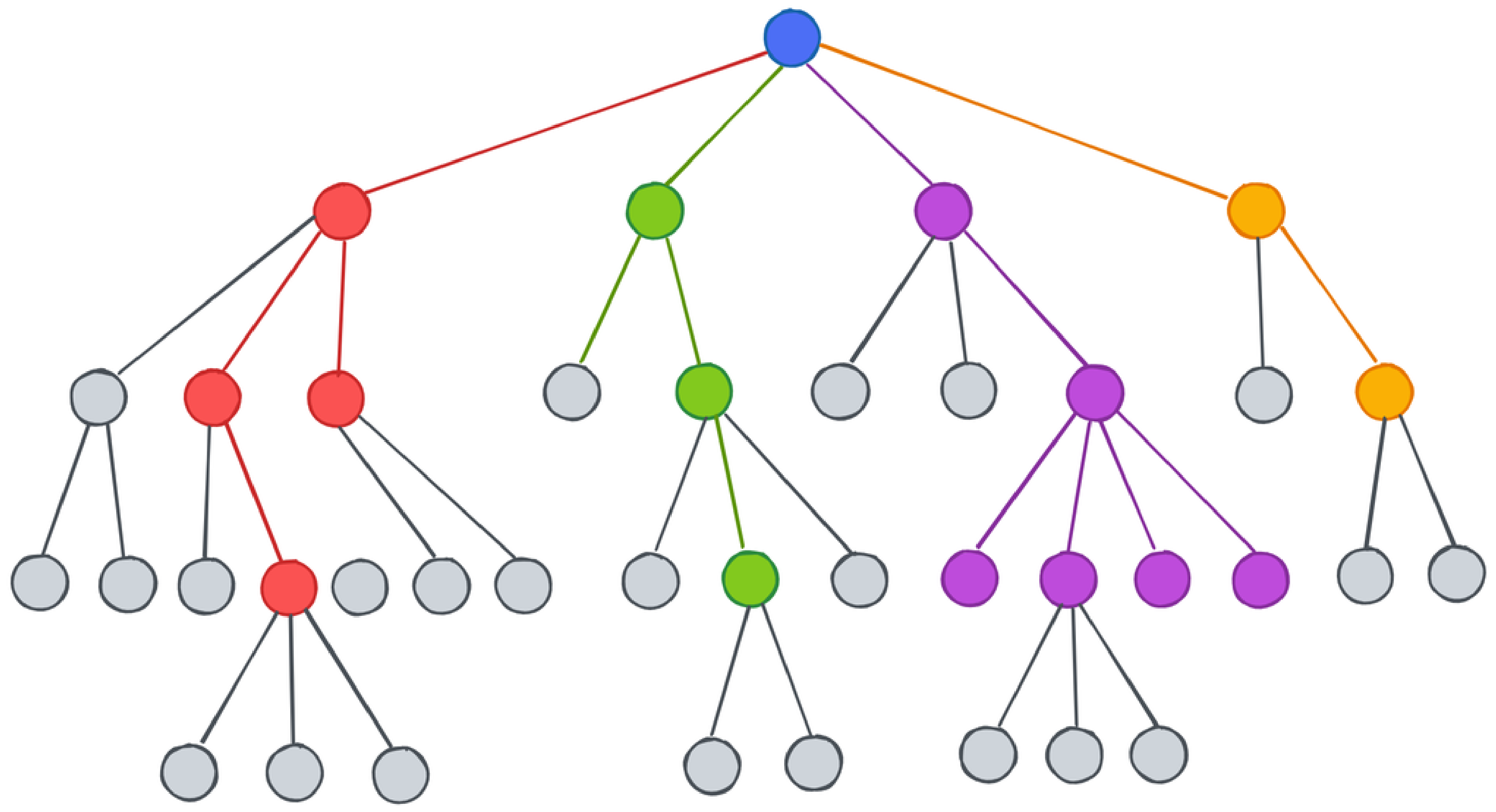
- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

DECISION TREE



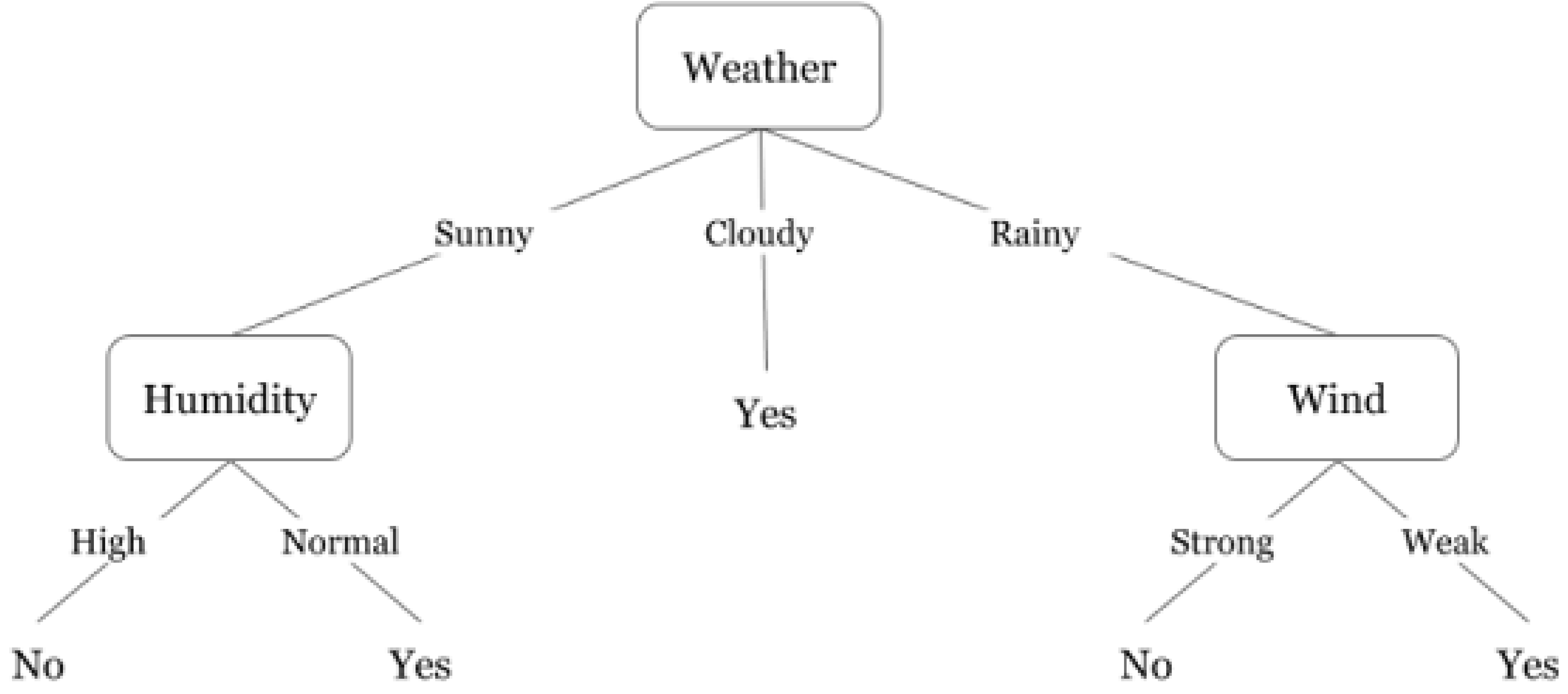


DECISION TREE

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

DECISION TREE

Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No



DECISION TREE

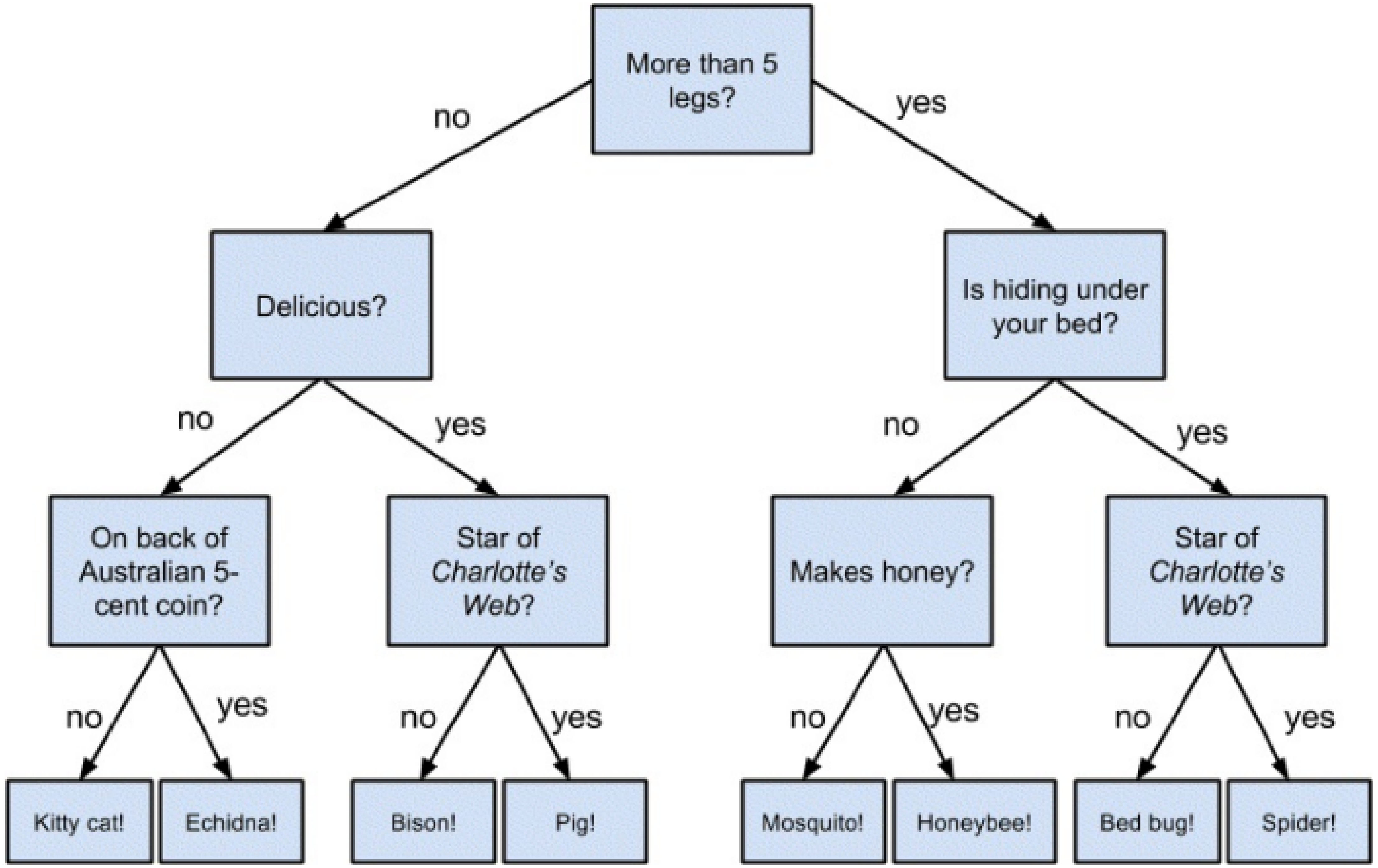


Figure 17-1. A "guess the animal" decision tree

ENTROPY

- In order to build a decision tree, we will need to decide what questions to ask and in what order.
- "Entropy" is a concept used to measure the impurity or disorder of a set of data points within a node of the decision tree.
- When building a decision tree, you want to minimize entropy.
- The idea is to find attribute values or features that, when used for splitting, result in nodes with lower entropy.

ENTROPY CALCULATION

In math terms, if p_i is the proportion of data labeled as class c_i, we define the entropy as:

$$H(S) = -p_1 \log_2 p_1 - \ldots - p_n \log_2 p_n$$

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

INFORMATION GAIN

- Information Gain is another important concept related to entropy in decision trees. It quantifies the reduction in entropy achieved by a particular split.

- The formula for Information Gain is:

  Information Gain = Entropy(parent) - Weighted Average of Entropy(children)

- Decision tree algorithms typically select the split with the highest Information Gain, as it represents the most effective way to reduce uncertainty or impurity in the data.

INFORMATION GAIN

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

parent entropy $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=$ **0.996**

child 1 entropy $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=$ **0.787**

child 2 entropy $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=$ **0.391**

(Weighted) Average Entropy of children **=** $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain - 0.996 - 0.615 = 0.38 for this split

COMPARISION OF GAIN VALUE

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

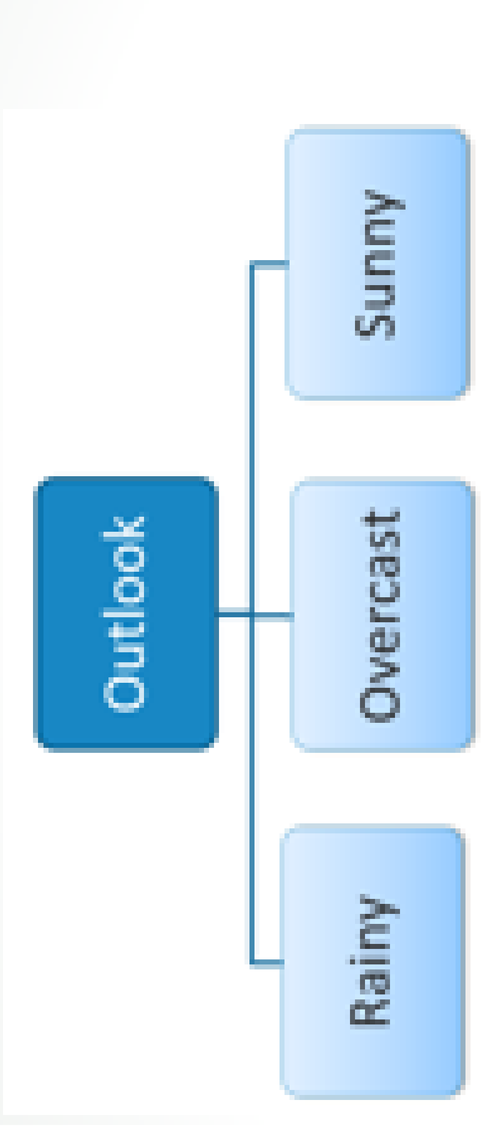
		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

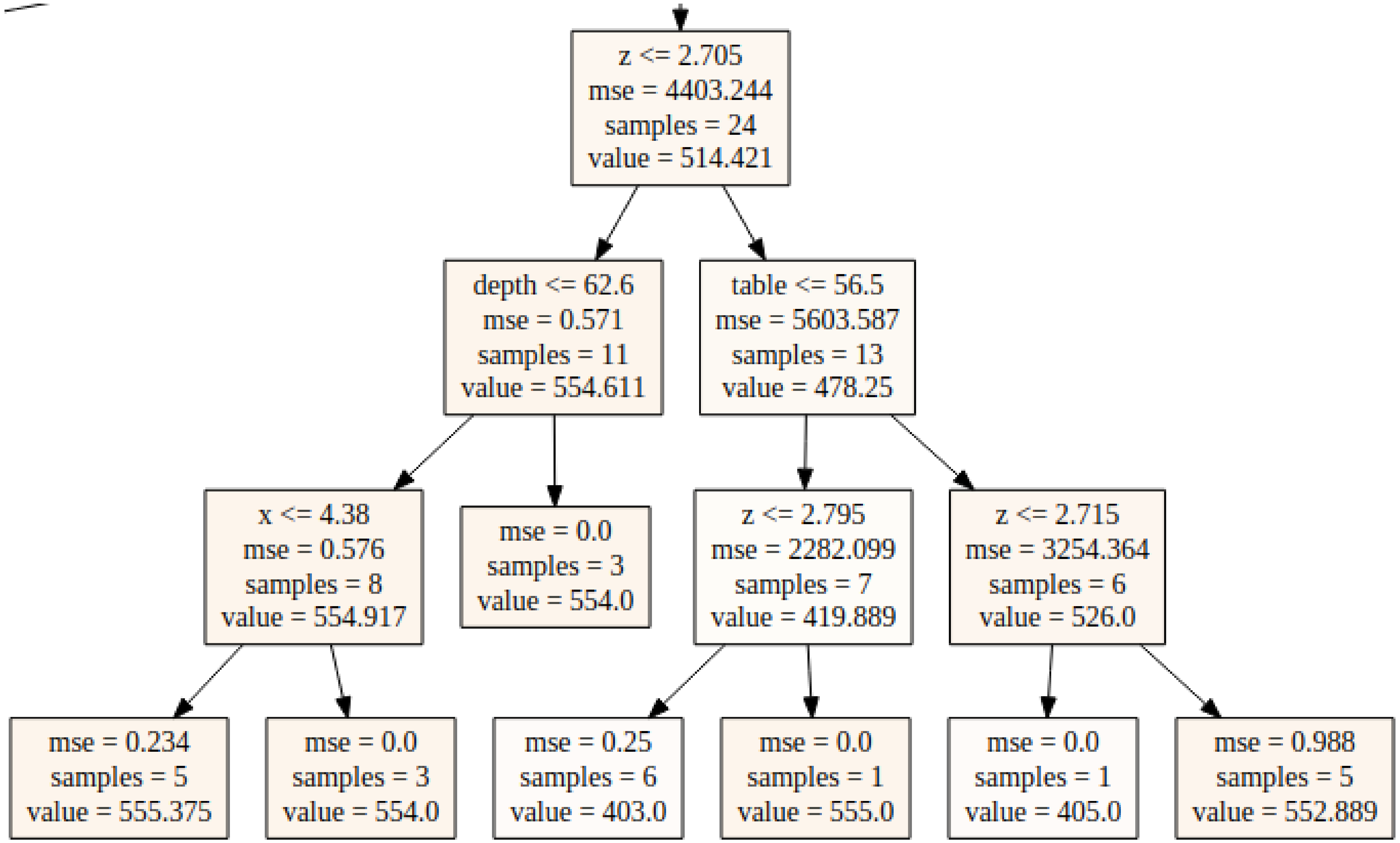
HIGHEST GAIN IS TAKEN

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

DECISION TREE REGRESSOR



DECISION TREE ALGORITHM

```
from sklearn.tree import DesicionTreeClassifier

from sklearn.tree import DesicionTreeRegressor
```

ENSEMBLE ALGORITHMS

1. Bagging - Random Forest
2. Boosting - AdaBoost, XGBoost

THANK YOU

Pantech e Learning

DIGITAL LEARNING SIMPLIFIED