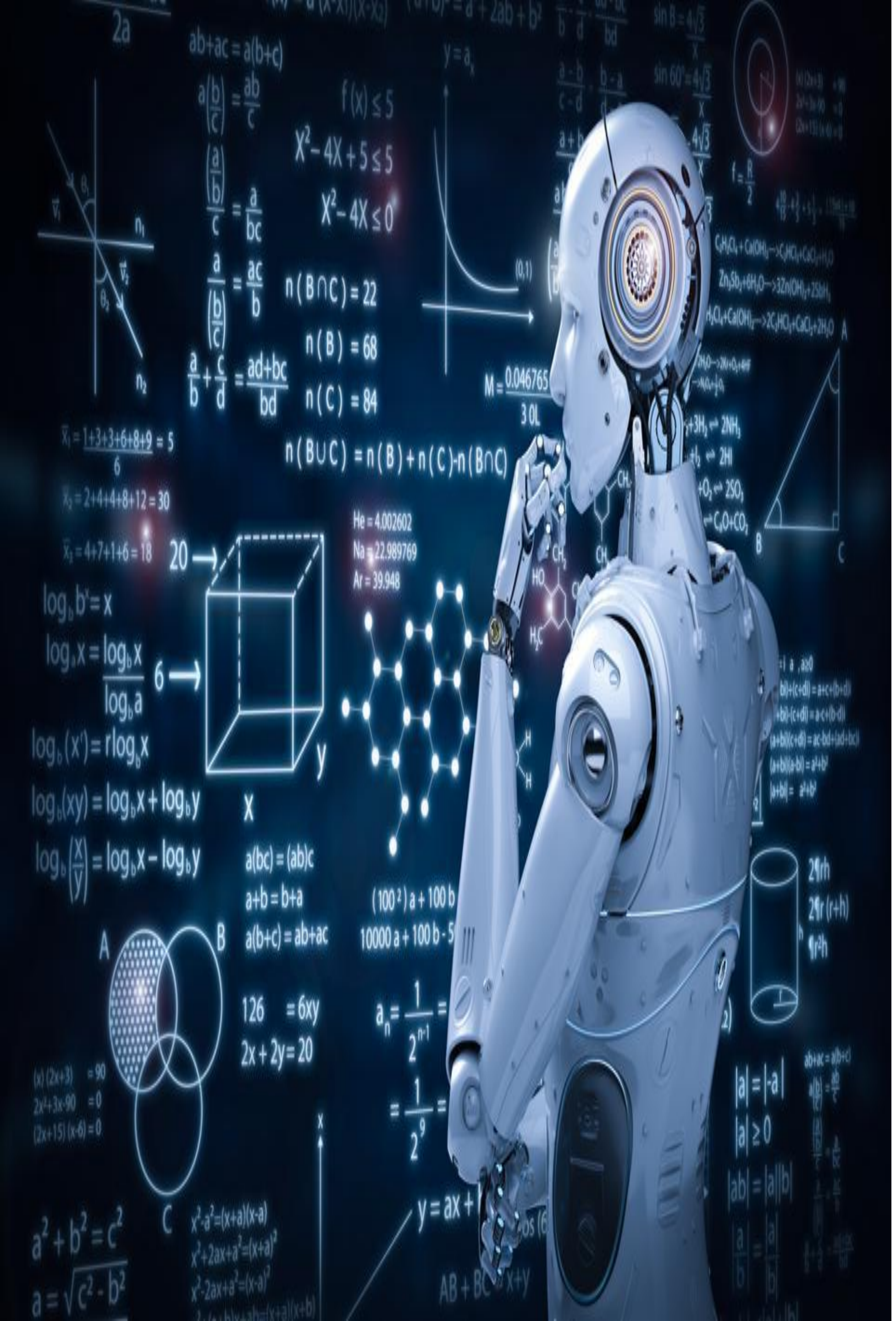


# Practical Machine Learning

## Day 13: Mar23 DBDA

Kiran Waghmare

# Agenda

- SVM
- SVM-Kernel

# SVM—Support Vector Machines

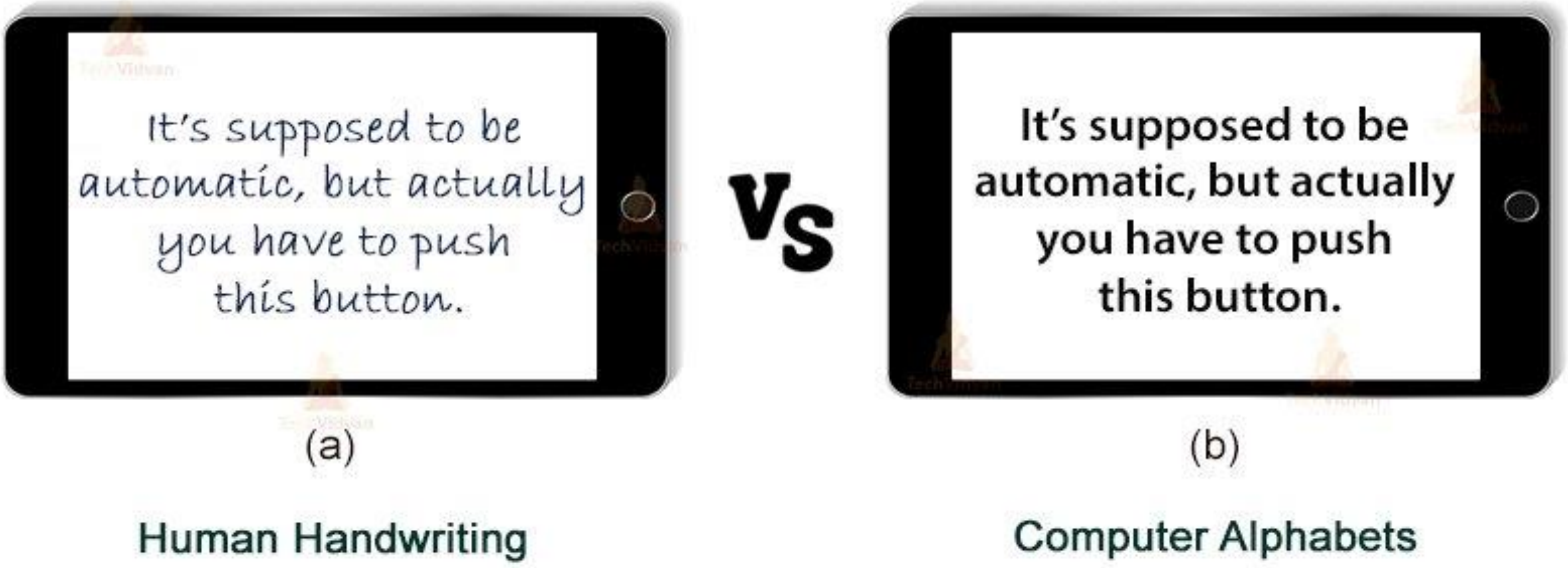
- A new classification method for both linear and nonlinear data
- It uses a nonlinear mapping to transform the original training data into a higher dimension
- With the new dimension, it searches for the linear optimal separating hyperplane (i.e., "decision boundary")
- With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane
- SVM finds this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors)

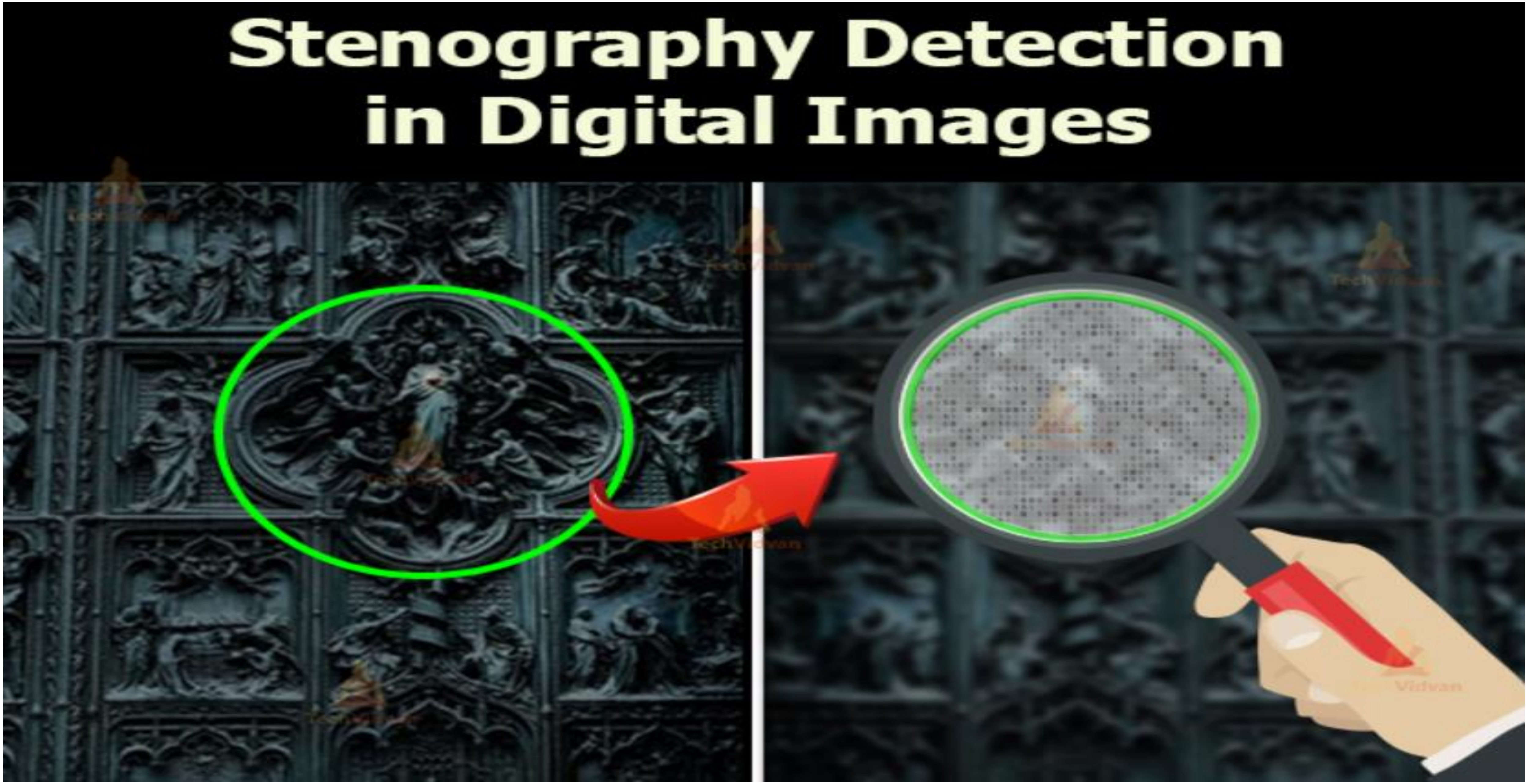
# Support Vector Machine Algorithm

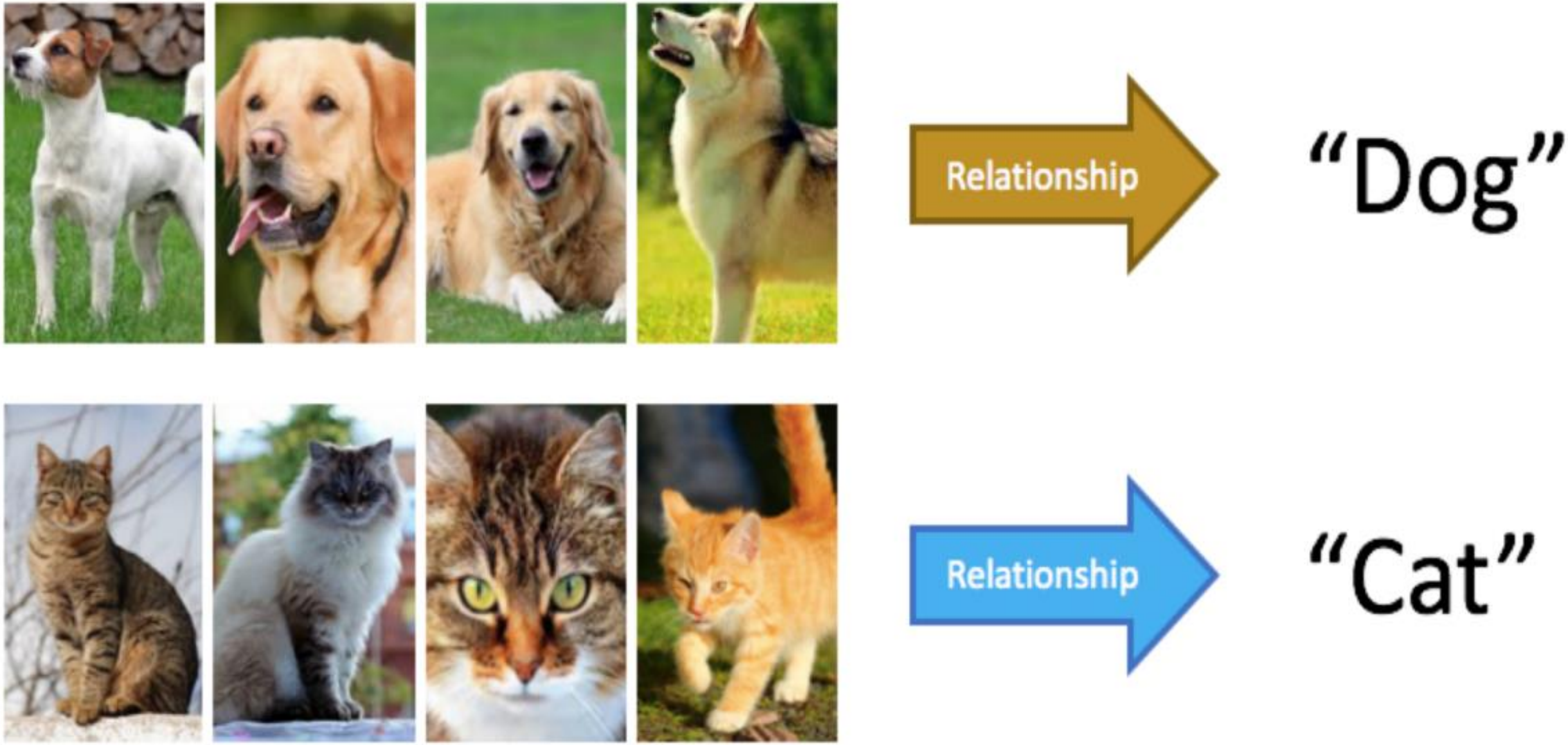
- **Goal :**
  - The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.
  - This best decision boundary is called **a hyperplane**
  - SVM chooses the extreme points/vectors that help in creating the hyperplane
  - These extreme cases are called as **support vectors**

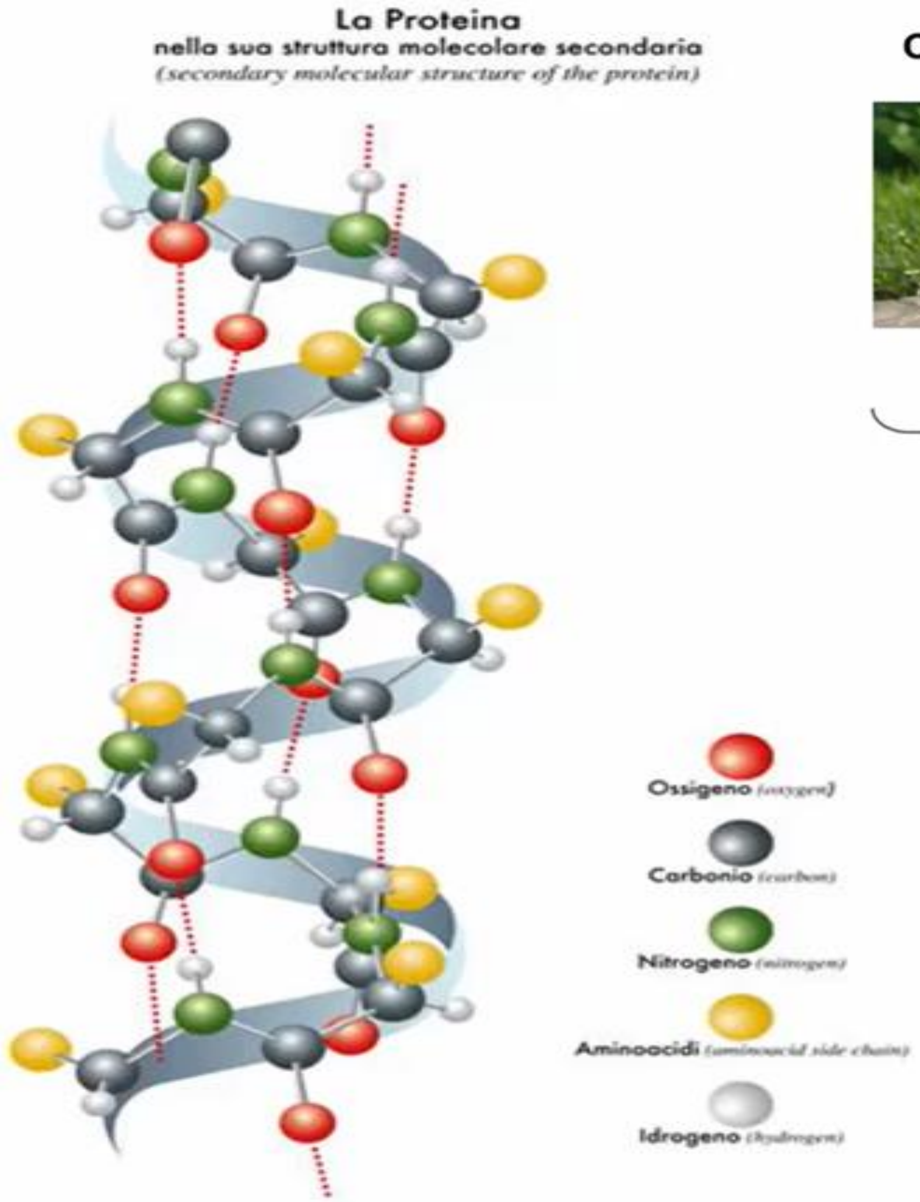
    and hence algorithm is termed as Support Vector Machine
- . Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

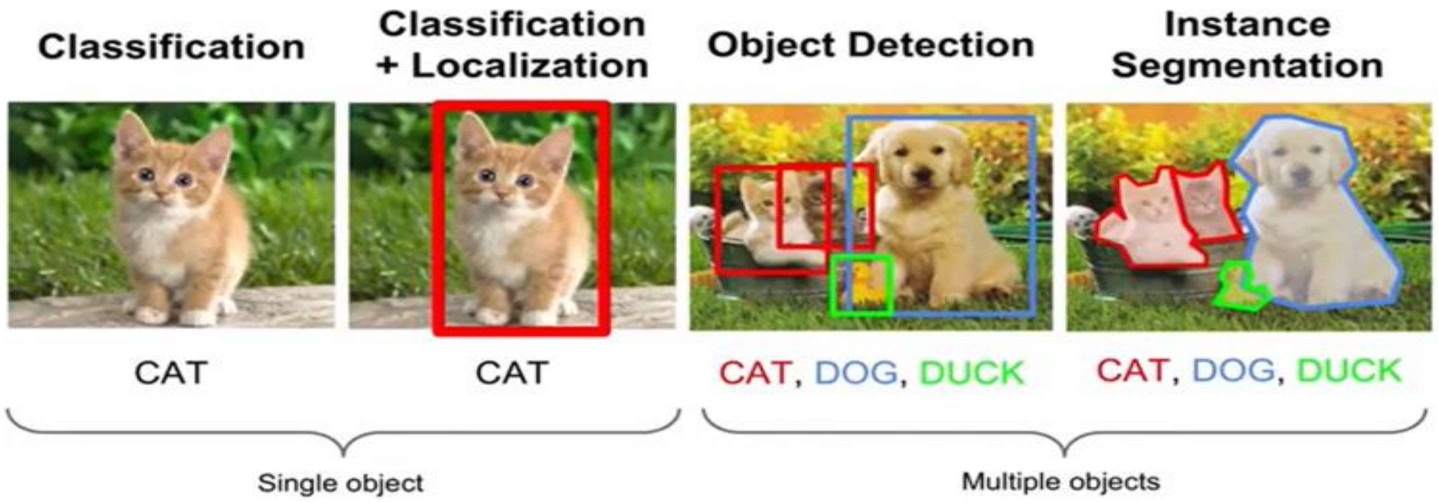
# Text Classification using SVM

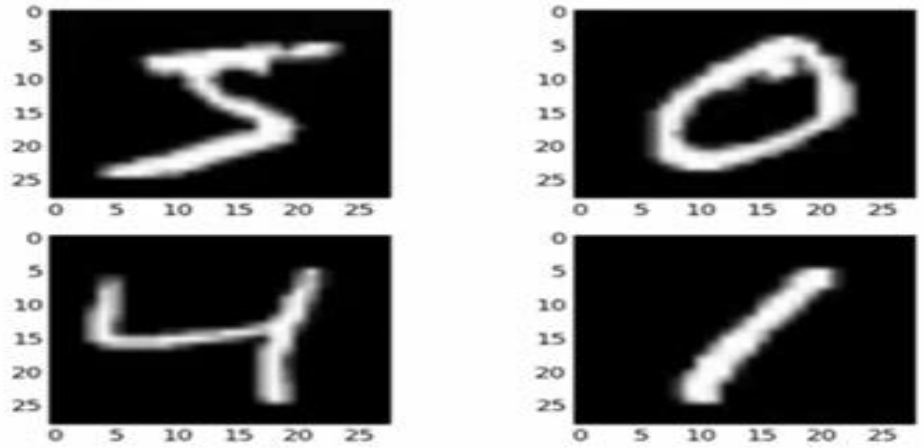



Stenography Detection
in Digital Images


Relationship
"Dog"
Relationship
"Cat"


La Proteina
nella sua struttura molecolare secondaria
(secondary molecular structure of the protein)
Ossigeno (oxygen)
Carbonio (carbon)
Nitrogeno (nitrogen)
Aminoacidi (aminoacid side chain)
Idrogeno (hydrogen)


Classification
Classification + Localization
Object Detection
Instance Segmentation
CAT
CAT
CAT, DOG, DUCK
CAT, DOG, DUCK
Single object
Multiple objects

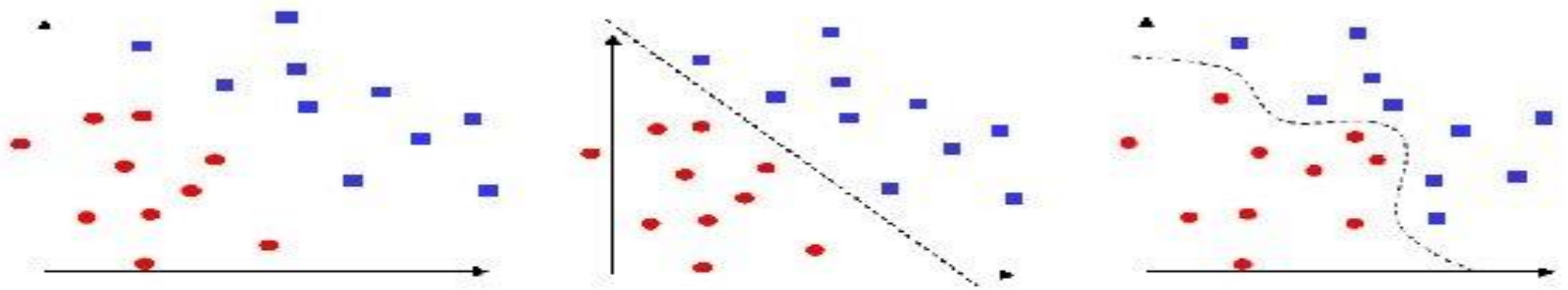


## Support Vector Machine (SVM)

-- classifier, forward neural network, supervised learning

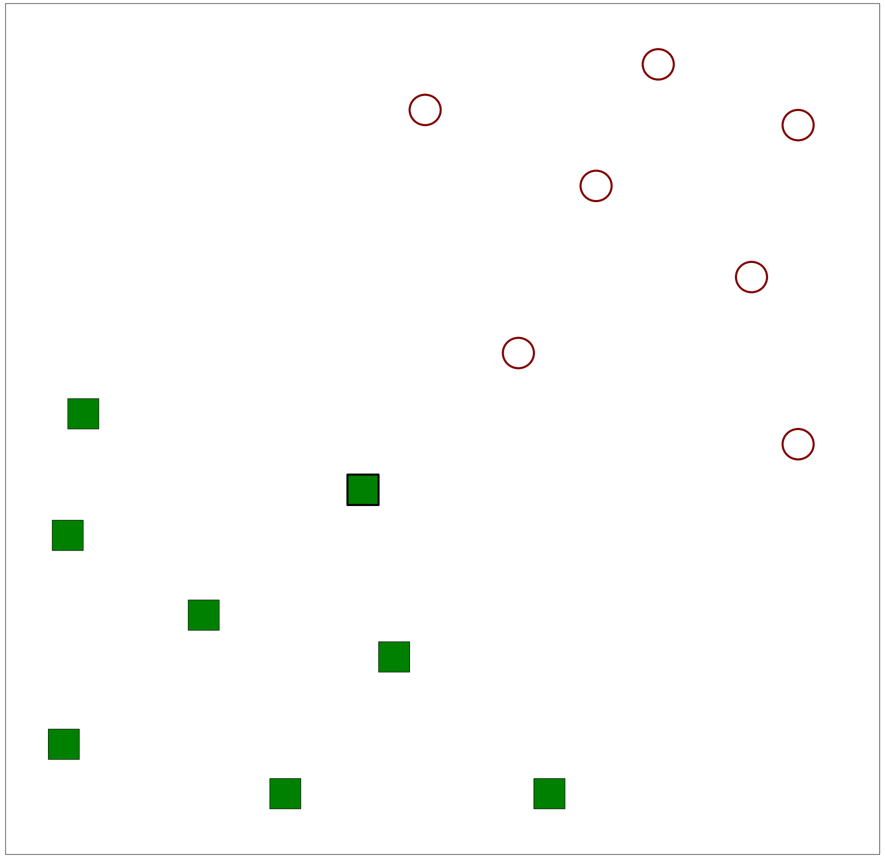
**Difficulties** with SVM:

i) binary classifier, ii) linearly separable patterns



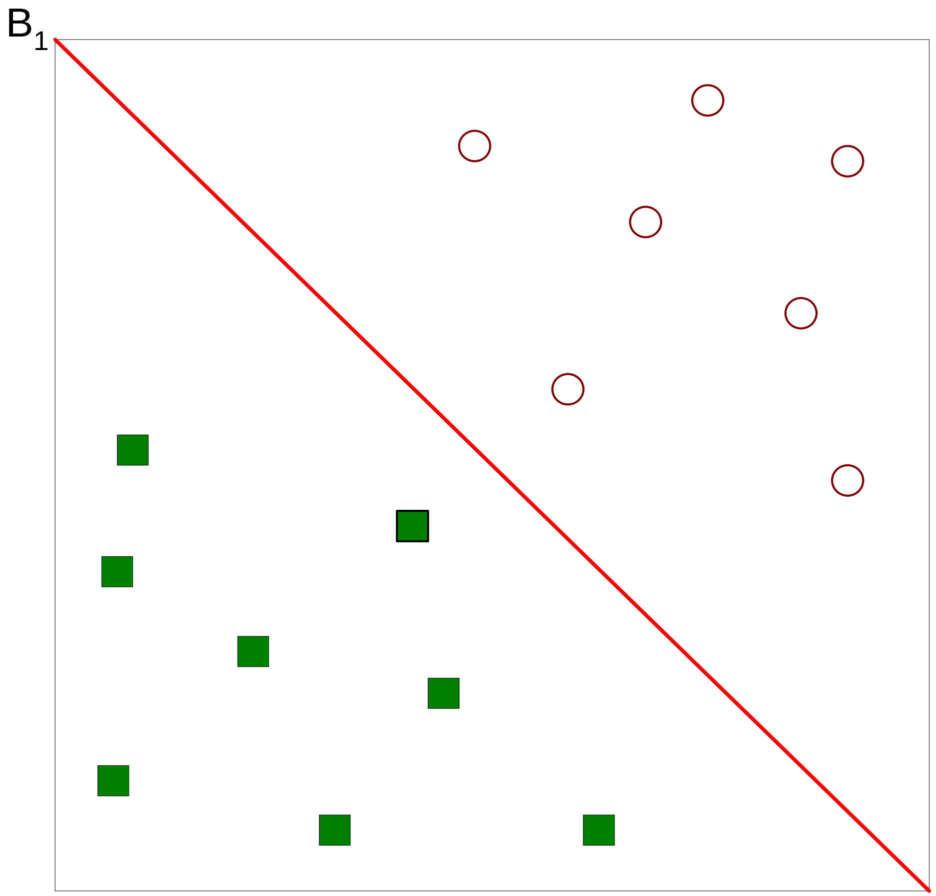
1

# Support Vector Machines



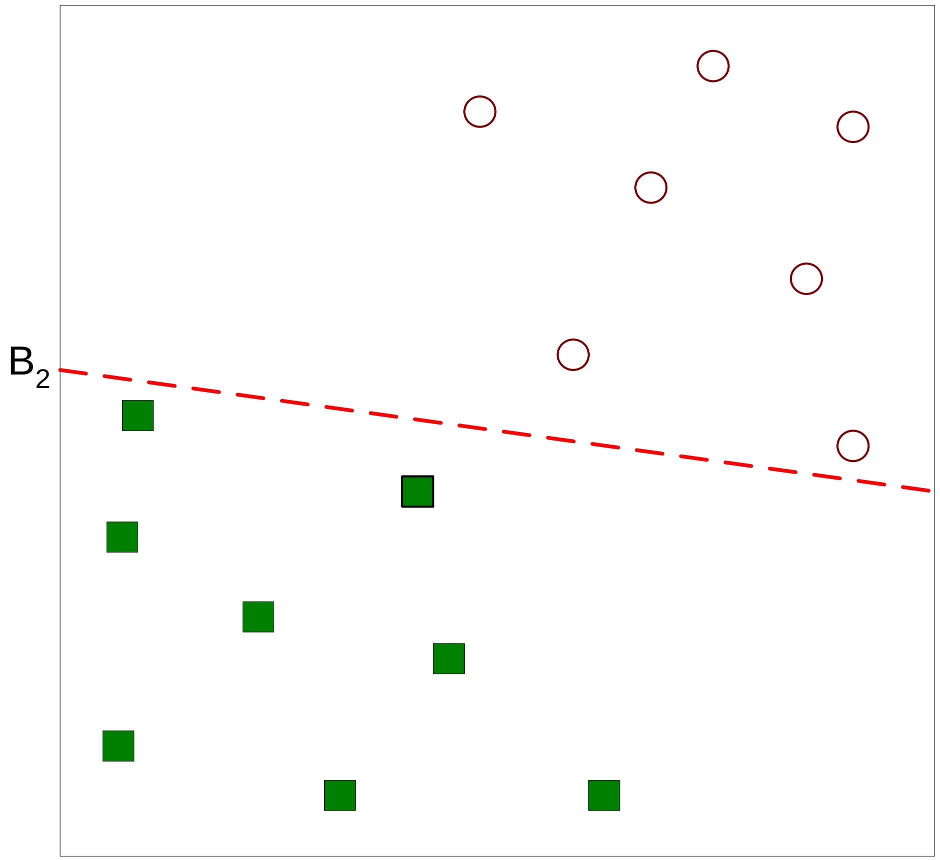
- Find a linear hyperplane (decision boundary) that will separate the data

# Support Vector Machines



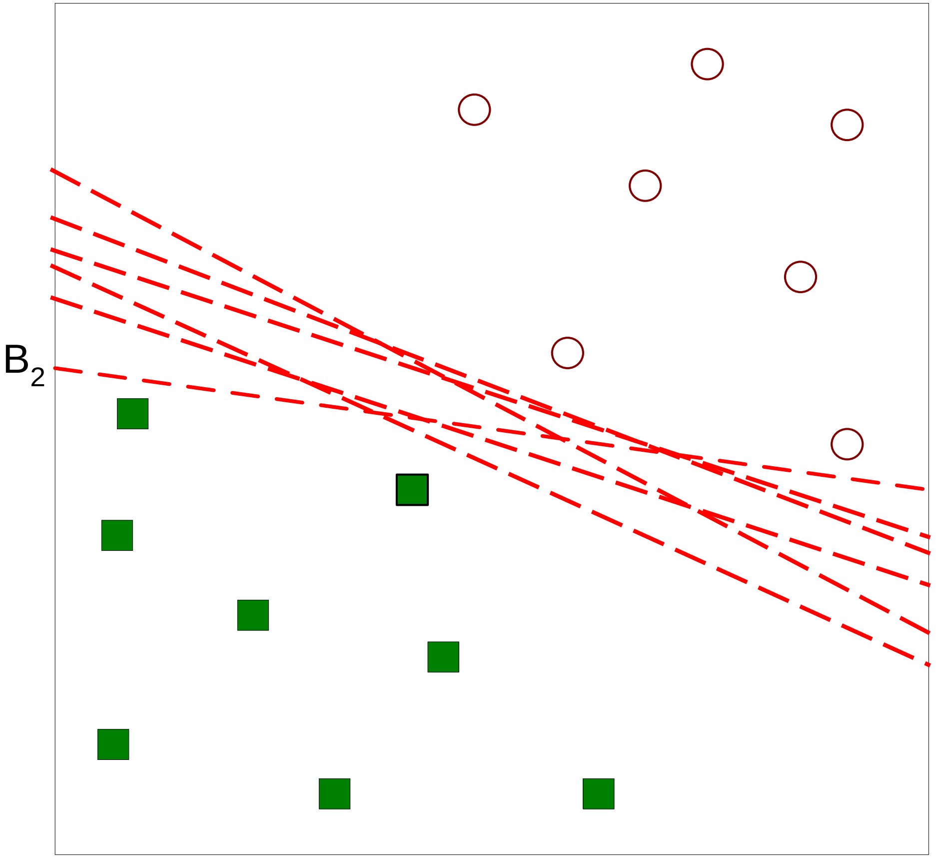

- One Possible Solution

# Support Vector Machines



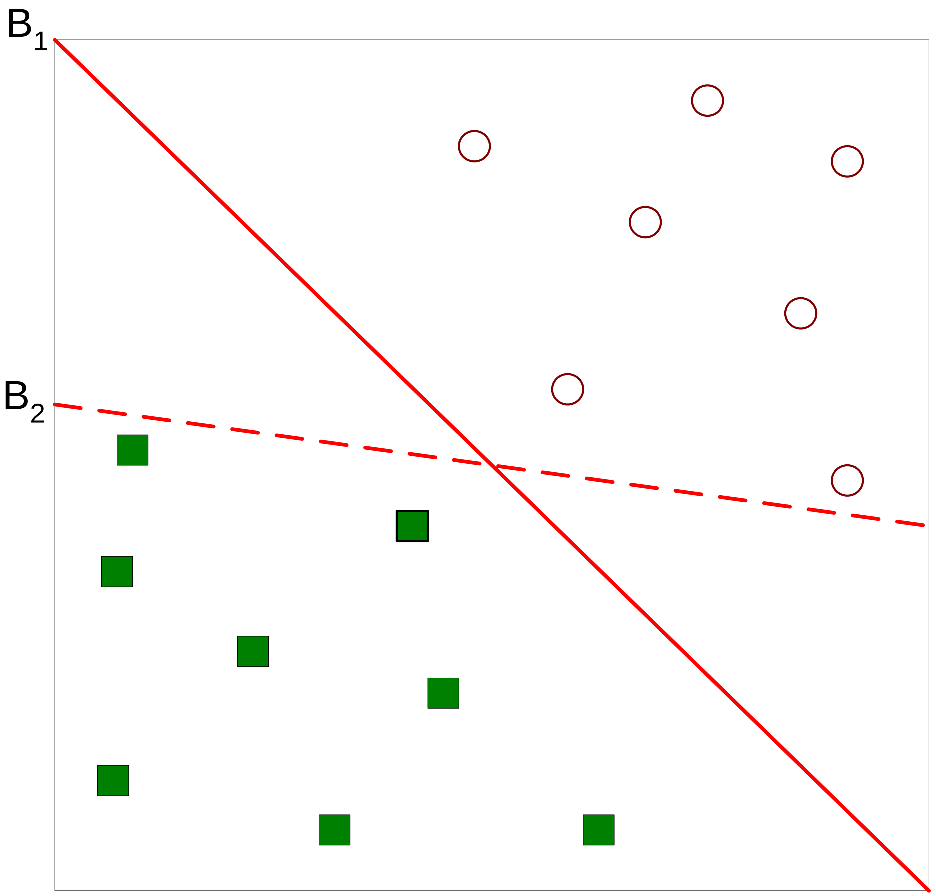

- Another possible solution

# Support Vector Machines



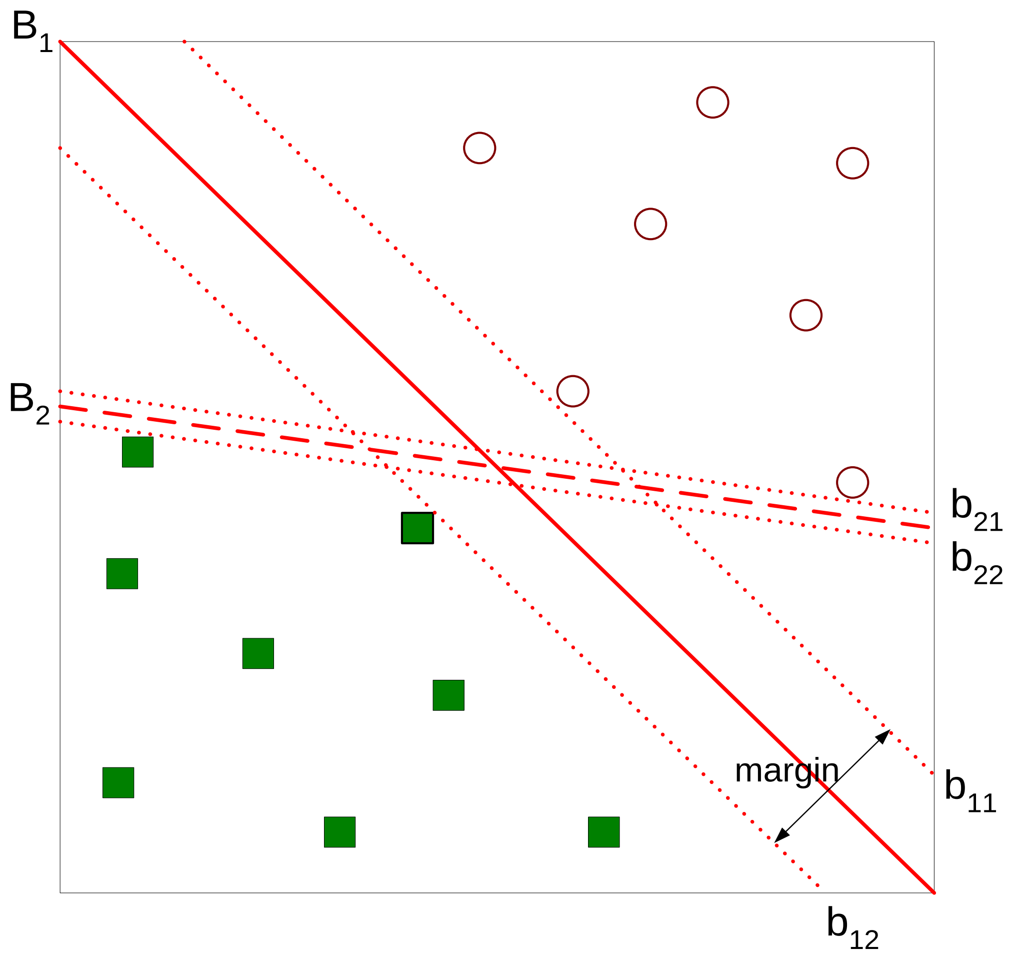

- Other possible solutions

# Support Vector Machines

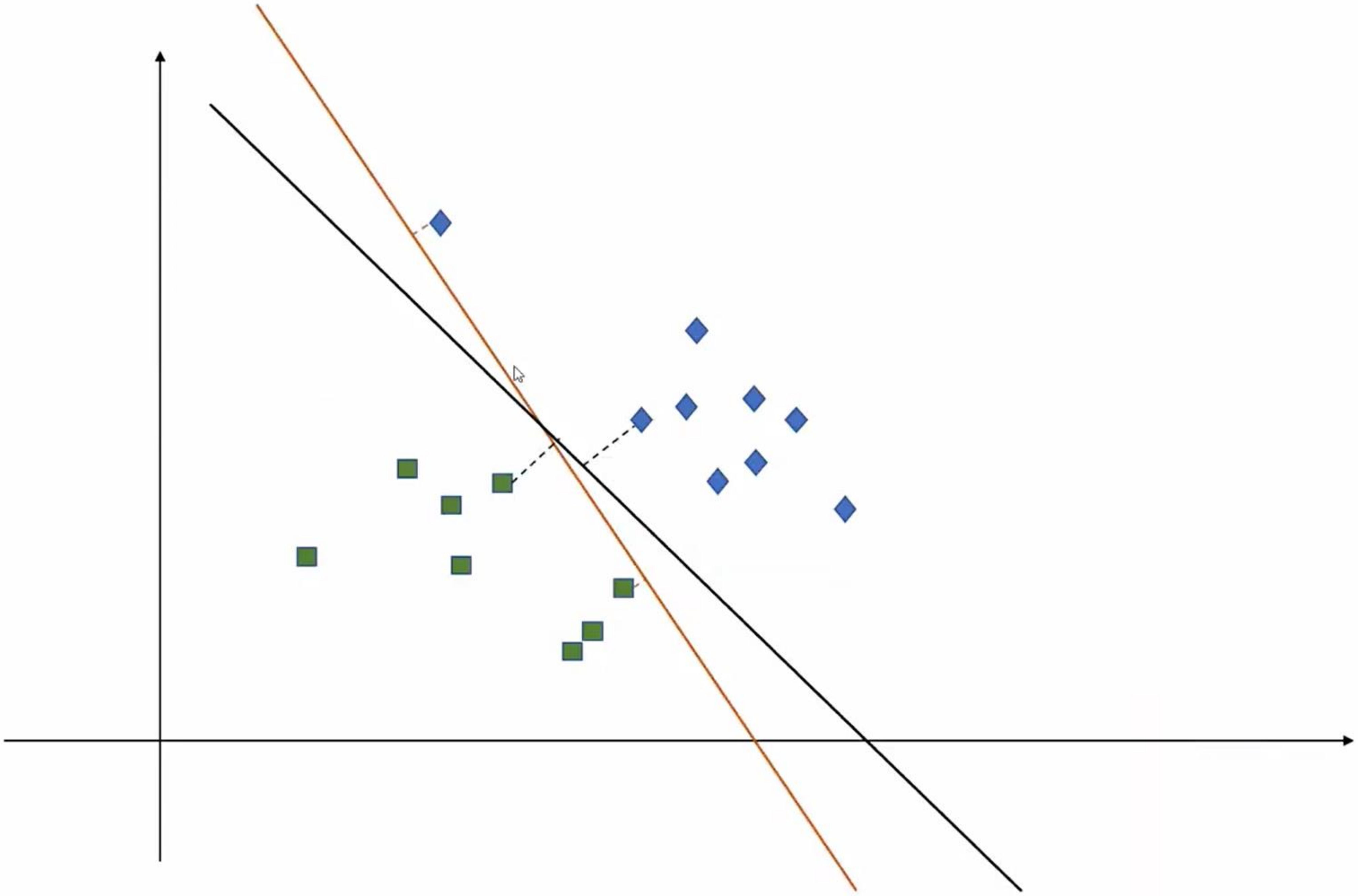


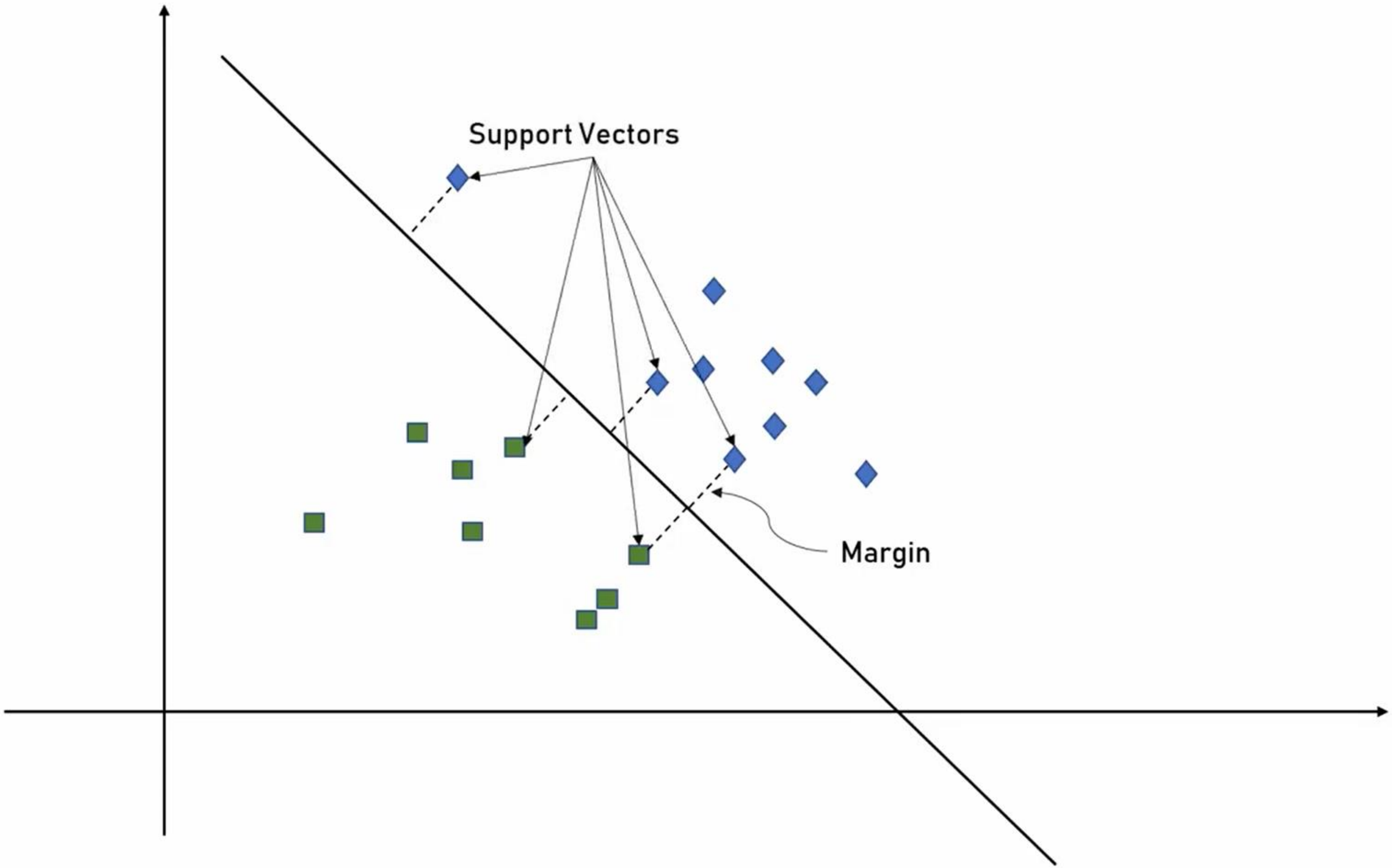

- Which one is better? B1 or B2?
- How do you define better?

# Support Vector Machines



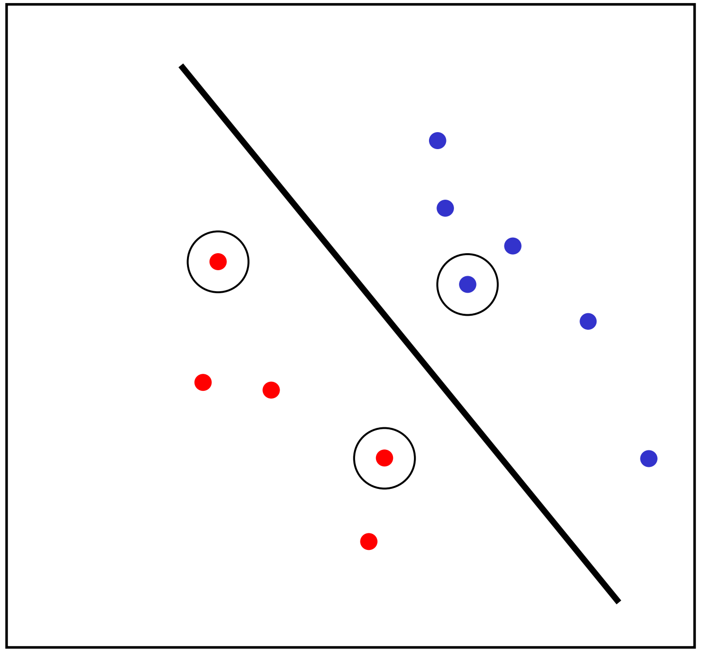

- Find hyperplane maximizes the margin => B1 is better than B2




Support Vectors
Margin

# Support Vector Machines

- The line that maximizes the minimum margin is a good bet.
  - The model class of “hyper-planes with a margin of m” has a low VC dimension if m is big.
- This maximum-margin separator is determined by a subset of the datapoints.
  - Datapoints in this subset  are called “support vectors”.

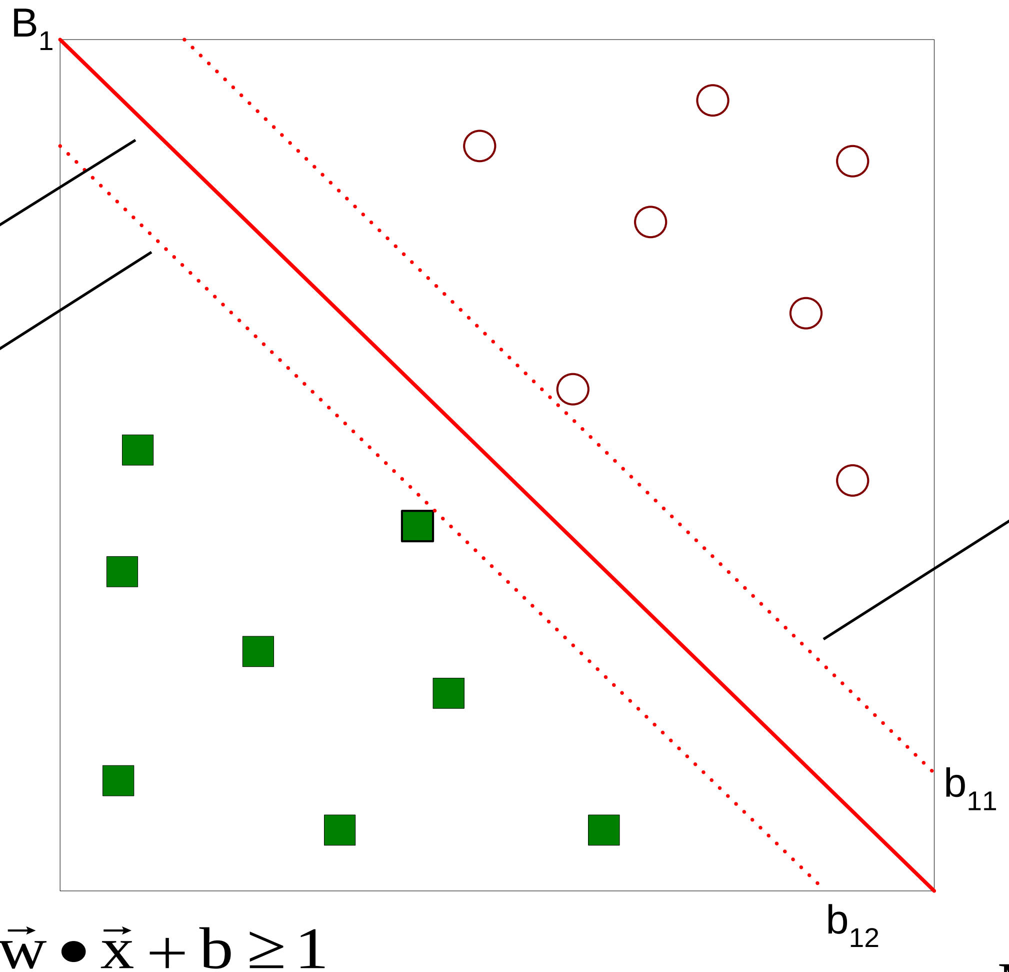


The support vectors are indicated by the circles around them.

# Support Vector Machines

$$\vec{w} \bullet \vec{x} + b = 0$$

$$\vec{w} \bullet \vec{x} + b = -1$$




$$\vec{w} \bullet \vec{x} + b = +1$$

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \leq -1 \end{cases}$$

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

# Training a linear SVM

- To find the maximum margin separator, we have to solve the following optimization problem:

$$\mathbf{w}.\mathbf{x}^c + b > +1 \quad \textit{for positive cases}$$

$$\mathbf{w}.\mathbf{x}^c + b < -1 \quad \textit{for negative cases}$$

$$\textit{and} \quad \| \mathbf{w} \|^2 \quad \textit{is as small as possible}$$

# Testing a linear SVM

- The separator is defined as the set of points for which:

$$\mathbf{w}.\mathbf{x} + b = 0$$

$$so\ if\ \ \mathbf{w}.\mathbf{x}^c + b > 0\ \ say\ its\ a\ \ positive\ \ case$$

$$and\ if\ \ \mathbf{w}.\mathbf{x}^c + b < 0\ \ say\ its\ a\ negative\ \ case$$

# A Bayesian Interpretation

- Using the maximum margin separator often gives a **pretty good approximation** to using all separators weighted by **their posterior probabilities.**

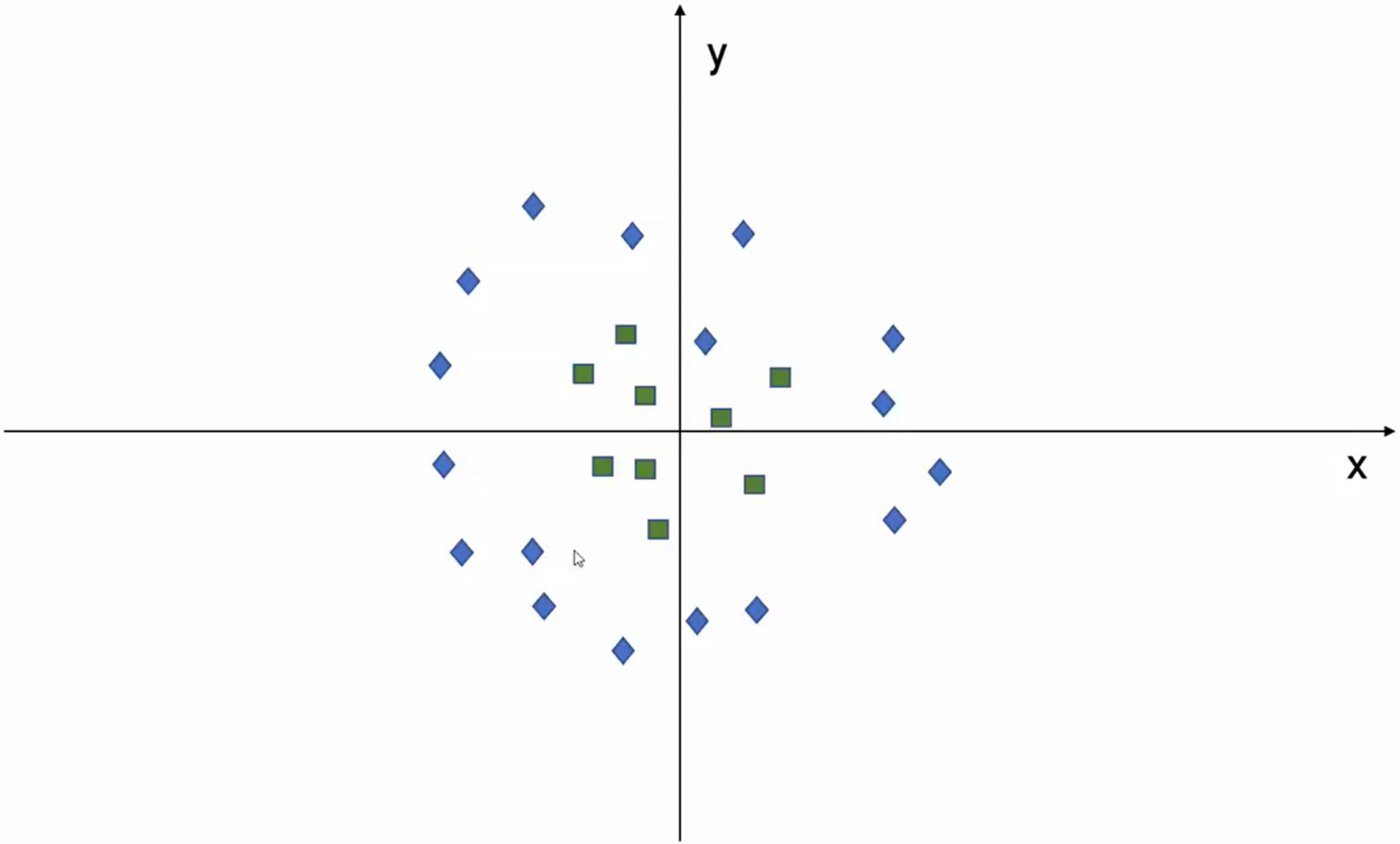
# What to do if there is no separating plane

- **Use a much bigger set of features.**
  - This looks as if it would make the computation hopelessly slow, but in the next part of the lecture we will see how to use the **"kernel"** trick to make the computation fast even with huge numbers of features.
- **Extend the definition of maximum margin to allow non-separating planes.**
  - This can be done by using "slack" variables

- **Kernel Function** is **a method used to take data as input and transform it into the required form of processing data**. “Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data.

- **What is kernel in machine learning?**
- In machine learning, a kernel refers to a method that allows us to apply linear classifiers to non-linear problems by mapping non-linear data into a higher-dimensional space without the need to visit or understand that higher-dimensional space.

- **What does kernel function do?**
- The kernel function is what is applied on each data instance to **map the original non-linear observations into a higher-dimensional space in which they become separable**.


y
x

# Some commonly used kernels

Polynomial: $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}.\mathbf{y} + 1)^p$

Gaussian radial basis function $K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$

Parameters that the user must choose

Neural net: $K(\mathbf{x},\mathbf{y}) = \tanh(k\,\mathbf{x}.\mathbf{y} - \delta)$

For the neural network kernel, there is one "hidden unit" per support vector, so the process of fitting the maximum margin hyperplane decides how many hidden units to use. Also, it may violate Mercer's condition.

# Introducing slack variables

- Slack variables are constrained to be non-negative. When they are greater than zero they allow us to cheat by putting the plane closer to the datapoint than the margin. So we need to minimize the amount of cheating. This means we have to pick a value for lamba (this sounds familiar!)

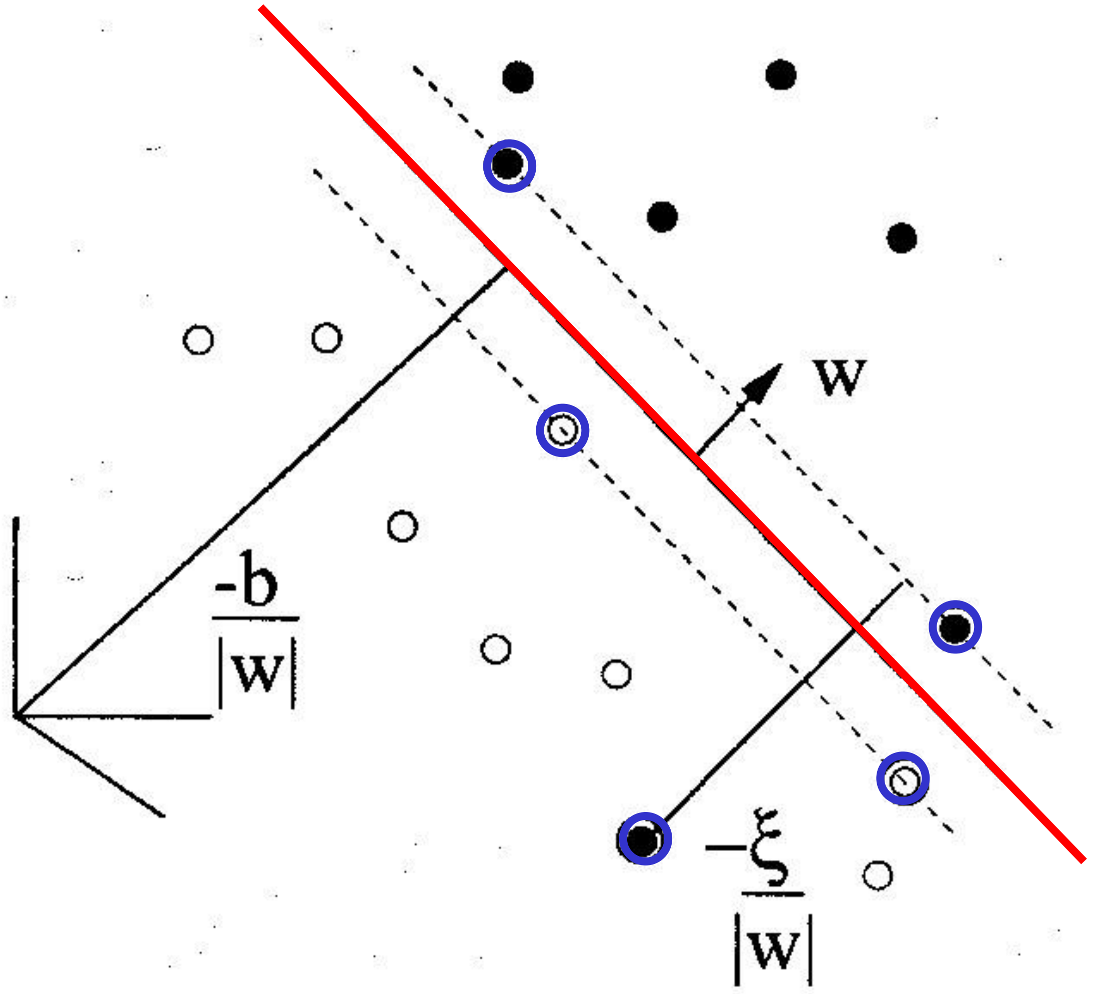
$$\mathbf{w}.\mathbf{x}^c + b \geq +1 - \xi^c \quad \textit{for positive cases}$$

$$\mathbf{w}.\mathbf{x}^c + b \leq -1 + \xi^c \quad \textit{for negative cases}$$

$$\textit{with} \quad \xi^c \geq 0 \quad \textit{for all } c$$

$$\textit{and} \quad \frac{\|\mathbf{w}\|^2}{2} + \lambda \sum_c \xi^c \quad \textit{as small as possible}$$

# A picture of the best plane with a slack variable



# Linear SVM

- Linear model:

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \leq -1 \end{cases}$$

- Learning the model is equivalent to determining the values of
  - How to find from training data?

$$\vec{w} \text{ and } b$$

$$\vec{w} \text{ and } b$$

# Learning Linear SVM

- Objective is to maximize:

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

  - Which is equivalent to minimizing:

$$L(\vec{w}) = \frac{\|\vec{w}\|^2}{2}$$

  - Subject to the following constraints:

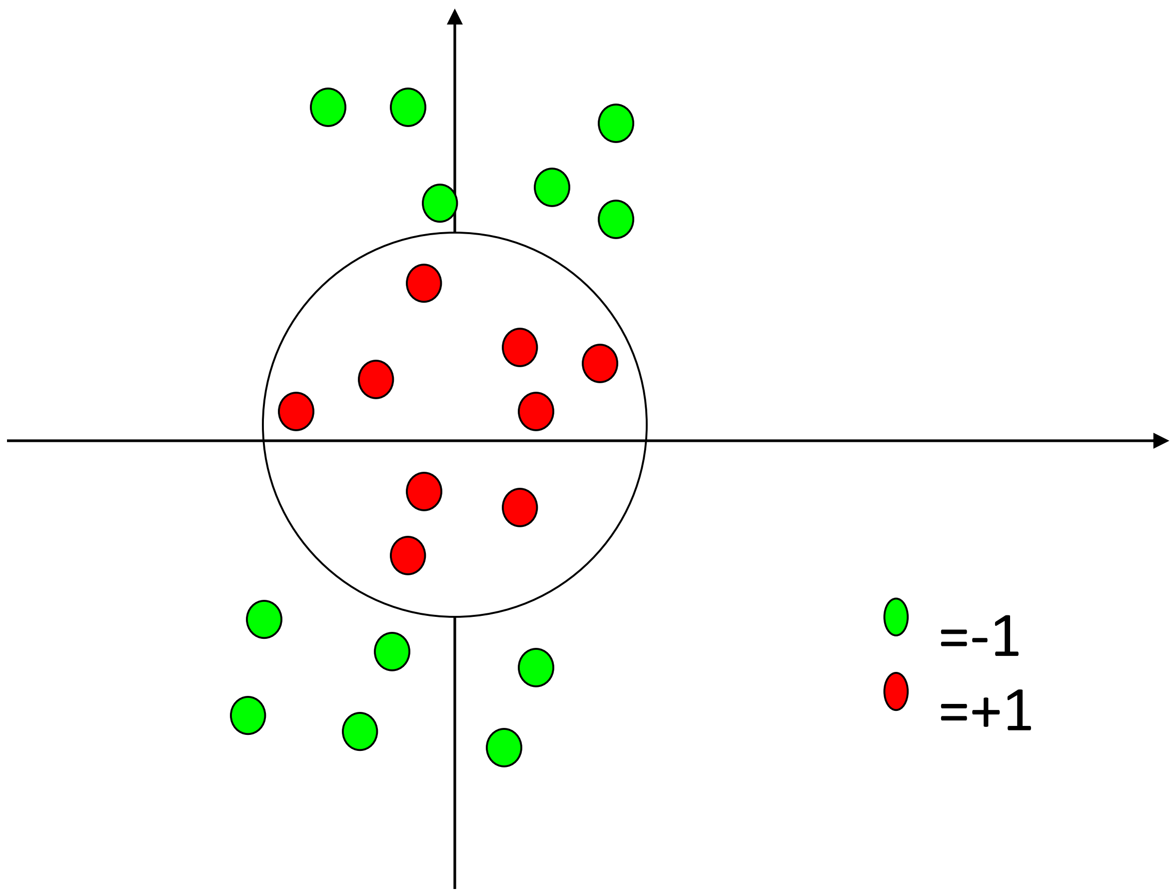
$$y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_{\mathrm{i}} + \mathrm{b} \leq -1 \end{cases}$$

or

$$y_i(\mathrm{w} \bullet \mathrm{x}_i + b) \geq 1, \qquad i = 1,2,\ldots,N$$

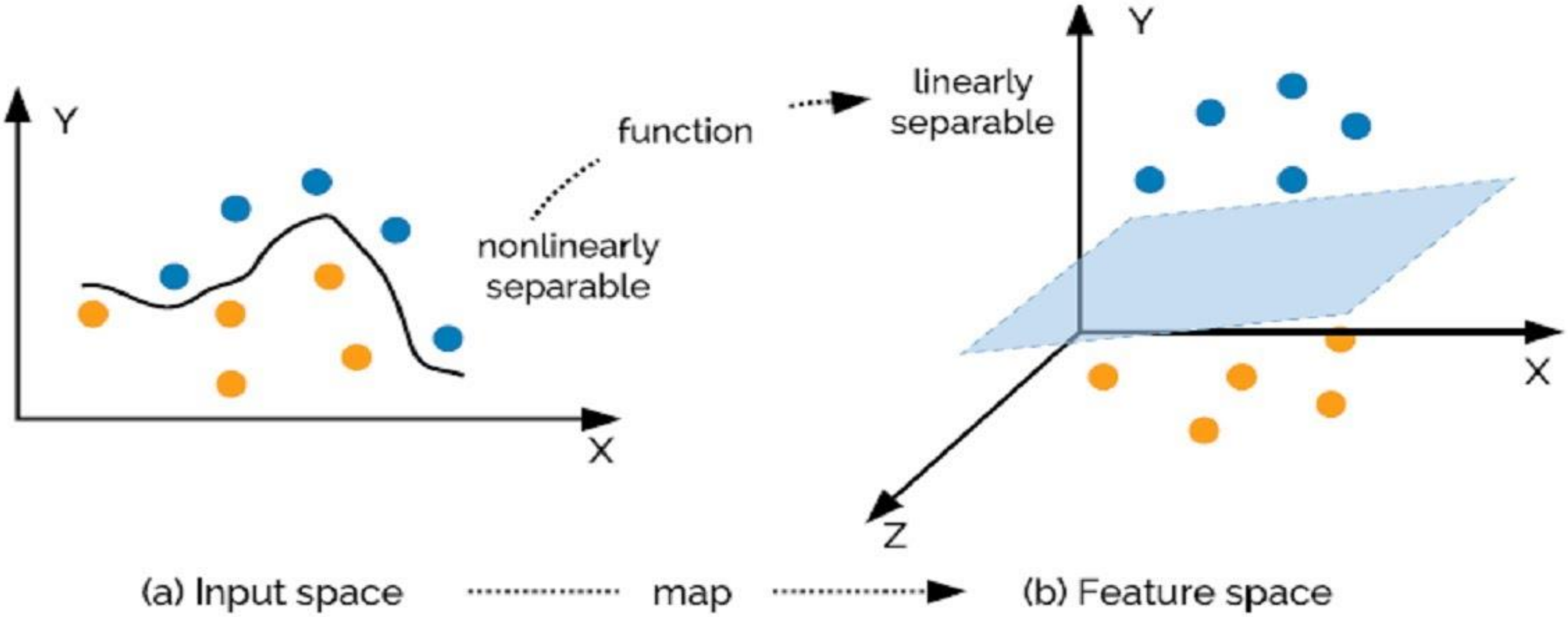
- This is a constrained optimization problem
  - Solve it using Lagrange multiplier method

# Problems with linear SVM



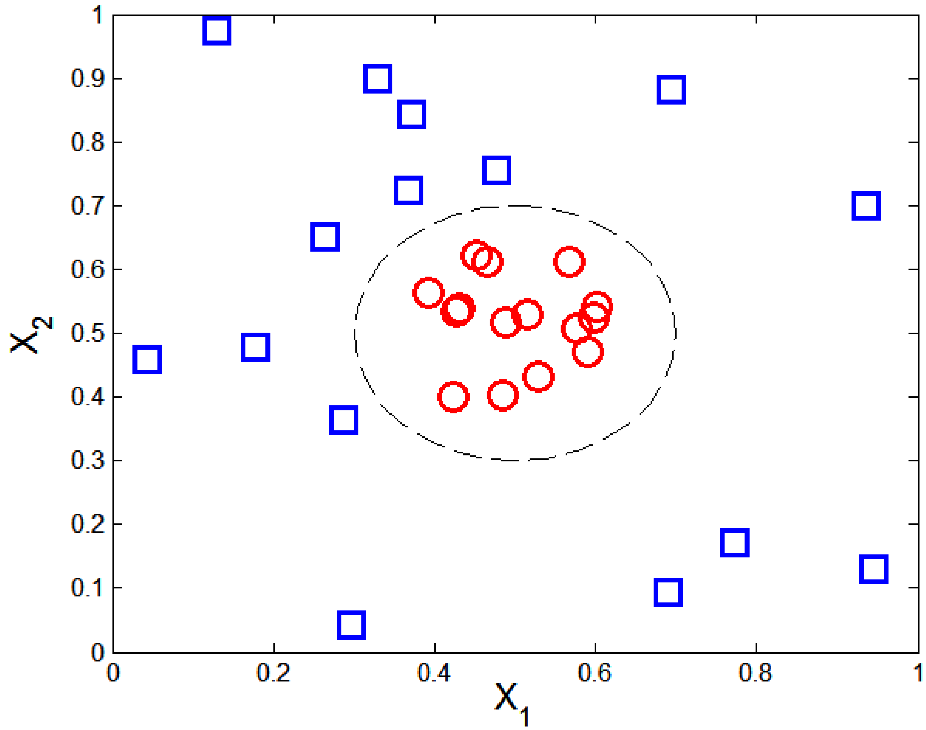

What if the decision function is not a linear?

# Kernal Trick (SVM)…



# Nonlinear Support Vector Machines

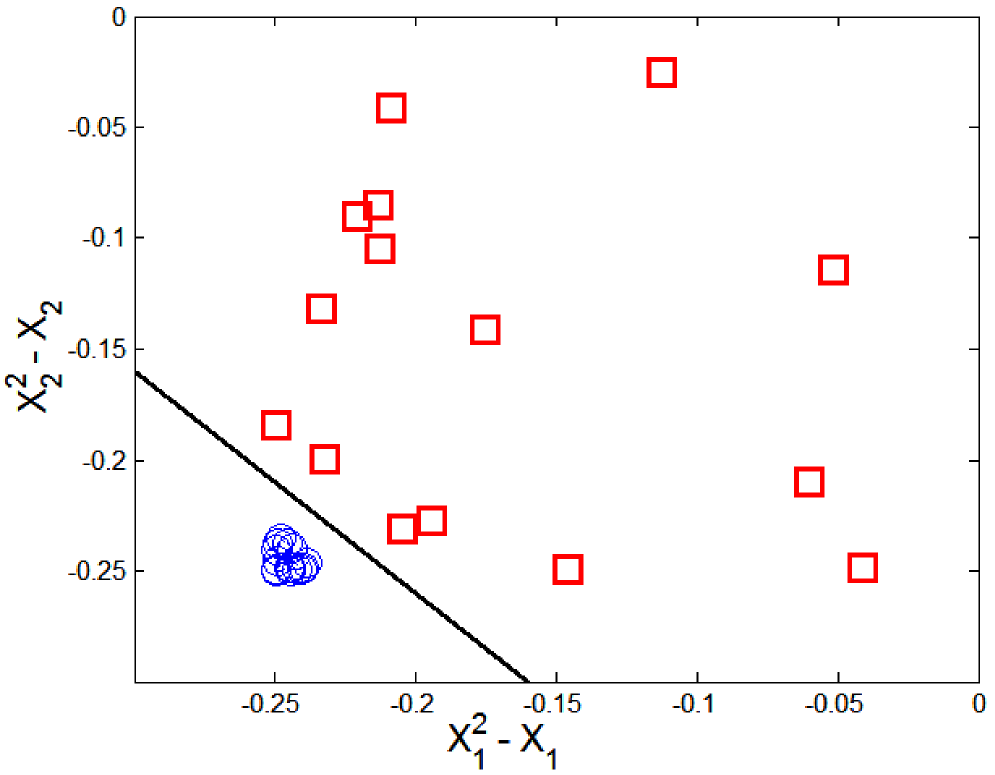
- What if decision boundary is not linear?




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

# Nonlinear Support Vector Machines

- Transform data into higher dimensional space




$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46.$$

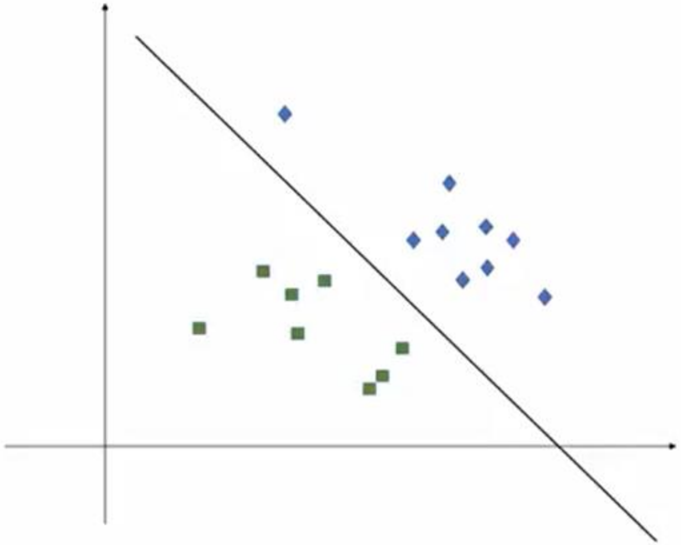
$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

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

Decision boundary:

$$\vec{w} \bullet \Phi(\vec{x}) + b = 0$$

2D



3D

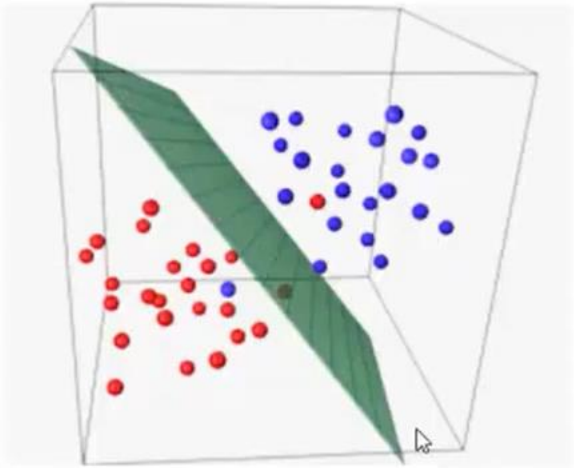
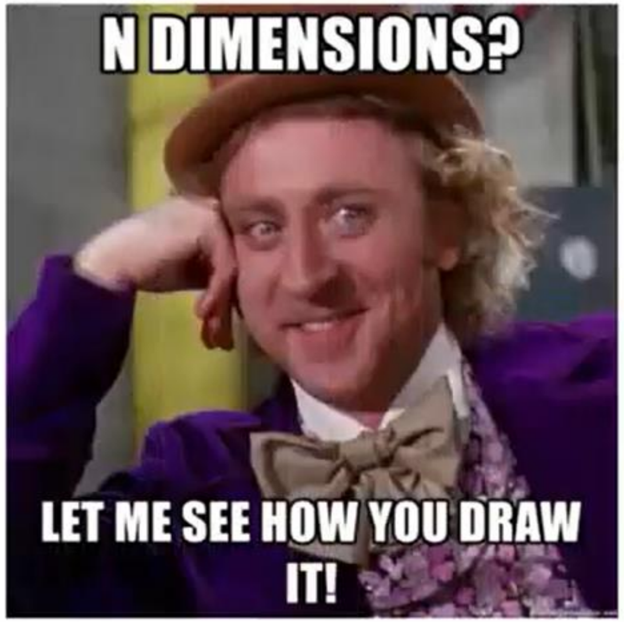
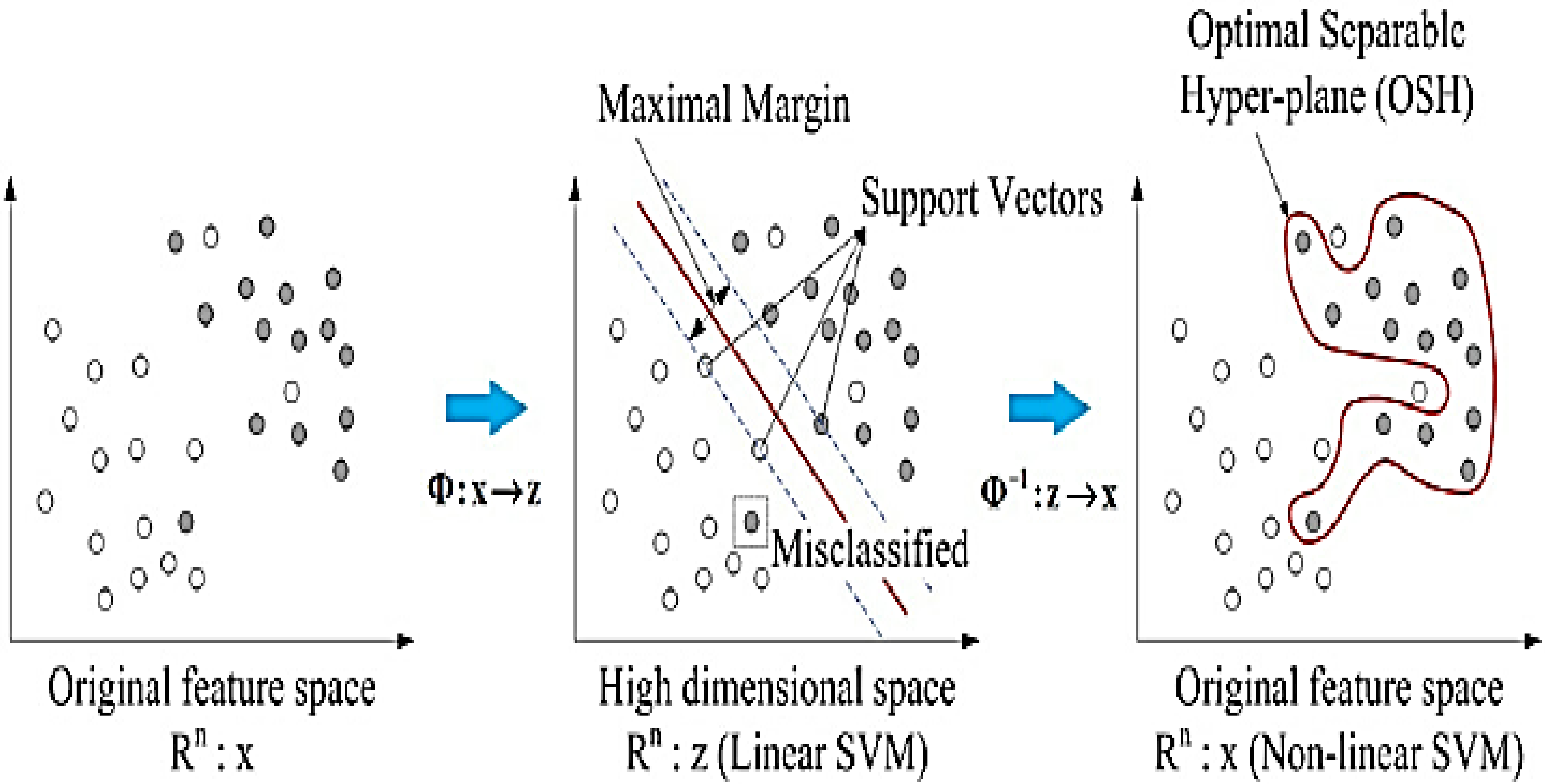


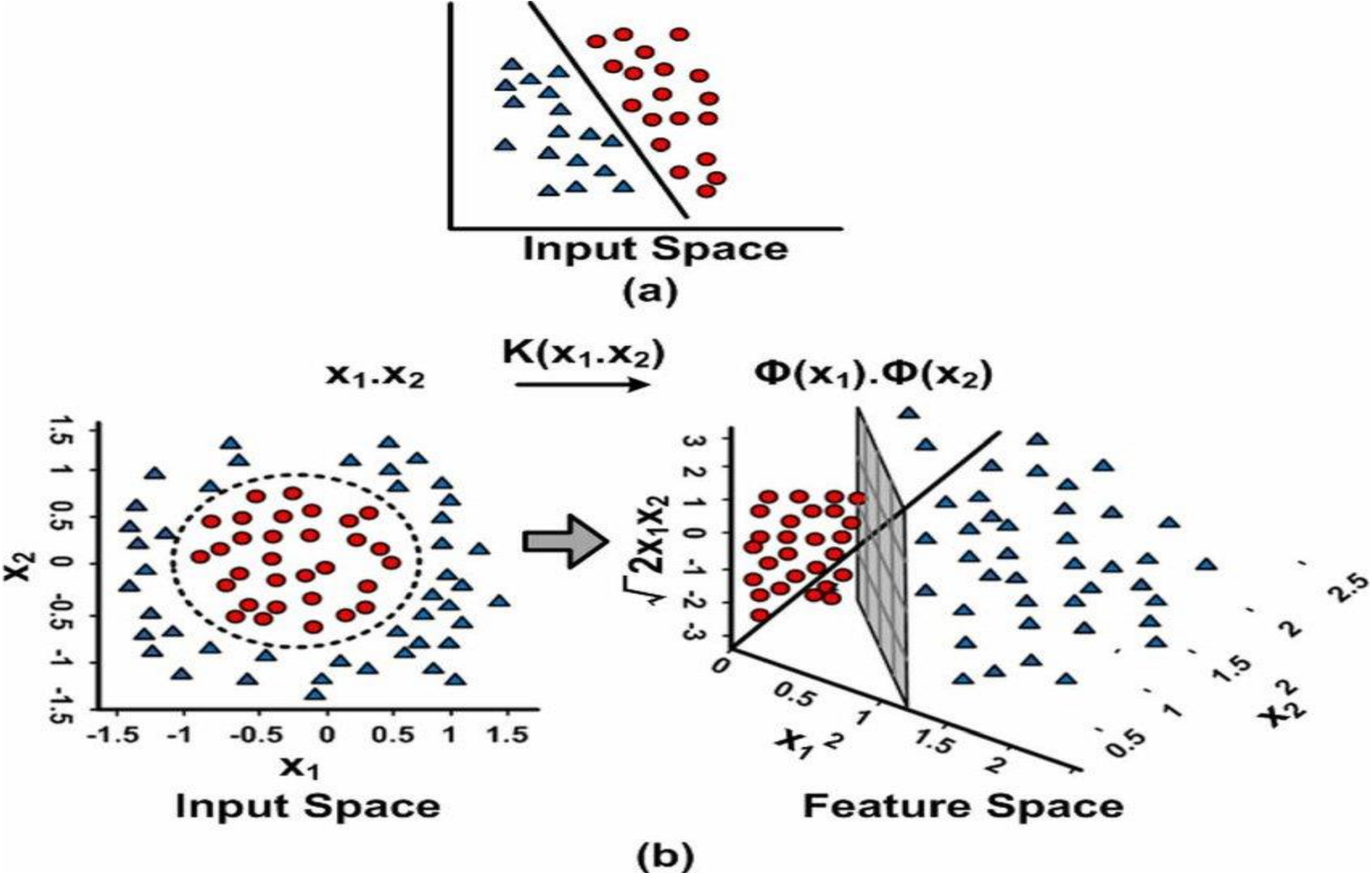
Image Credit: https://appliedmachinelearning.blog

nD




Optimal Separable Hyper-plane (OSH)
Maximal Margin
Support Vectors
Misclassified
$\Phi : x \to z$
$\Phi^{-1} : z \to x$
Original feature space
$R^n : x$
High dimensional space
$R^n : z$ (Linear SVM)
Original feature space
$R^n : x$ (Non-linear SVM)

Support vector machine draws a hyper plane in n dimensional space such that it maximizes margin between classification groups


Input Space
(a)
x1.x2
K(x1.x2)
Φ(x1).Φ(x2)
X2
1.5 1 0.5 0 -0.5 -1 -1.5
-1.5 -1 -0.5 0 0.5 1 1.5
X1
Input Space
√2x1x2
3 2 1 0 -1 -2 -3
0 0.5 1 1.5 2
X1
0.5 1 1.5 2 2.5
X2 2
Feature Space
(b)

# Overtraining/overfitting

A well known problem with machine learning methods is overtraining. This means that we have learned the training data very well, but we can not classify unseen examples correctly.

An example: A botanist really knowing trees. Everytime he sees a new tree, he claims it is not a tree.