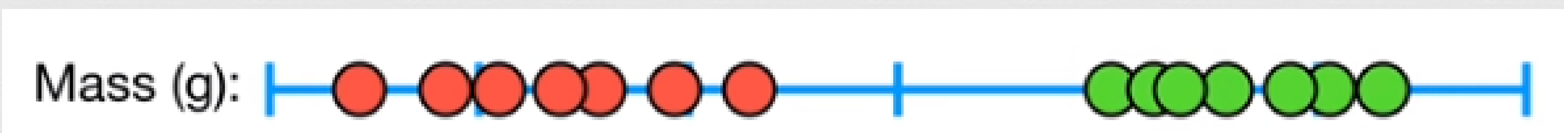
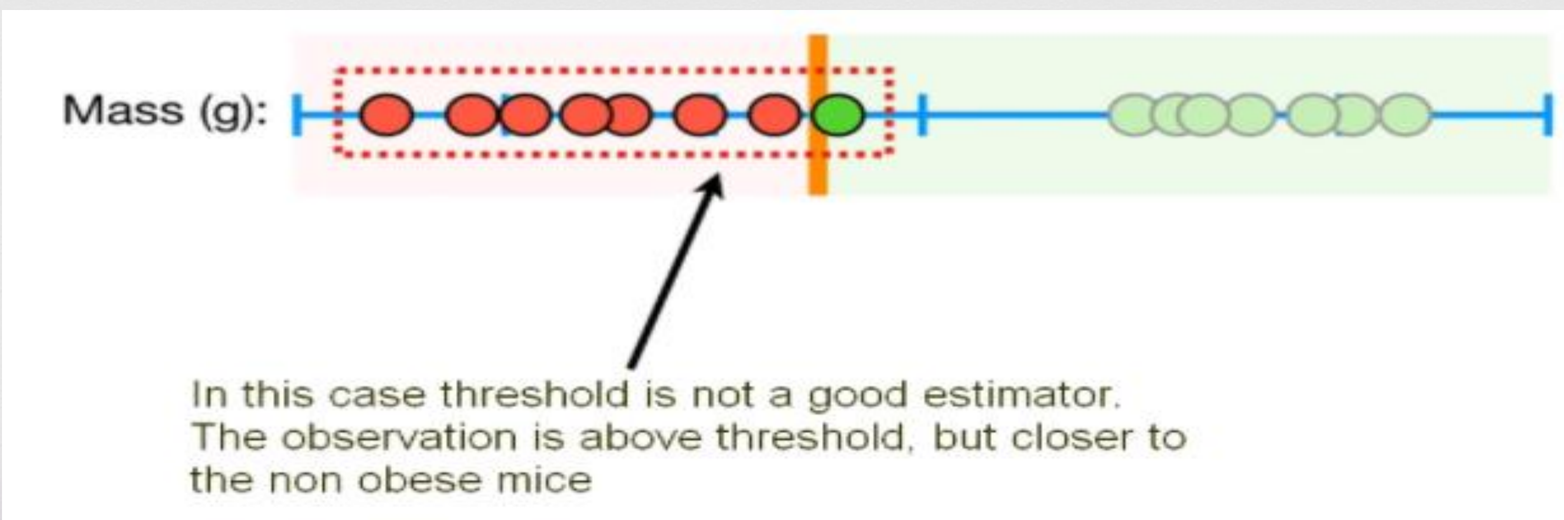
MCSC201: Machine Learning

Dr. Ankit Rajpal
Department of Computer Science
University of Delhi

Support Vector Machine: Intuition

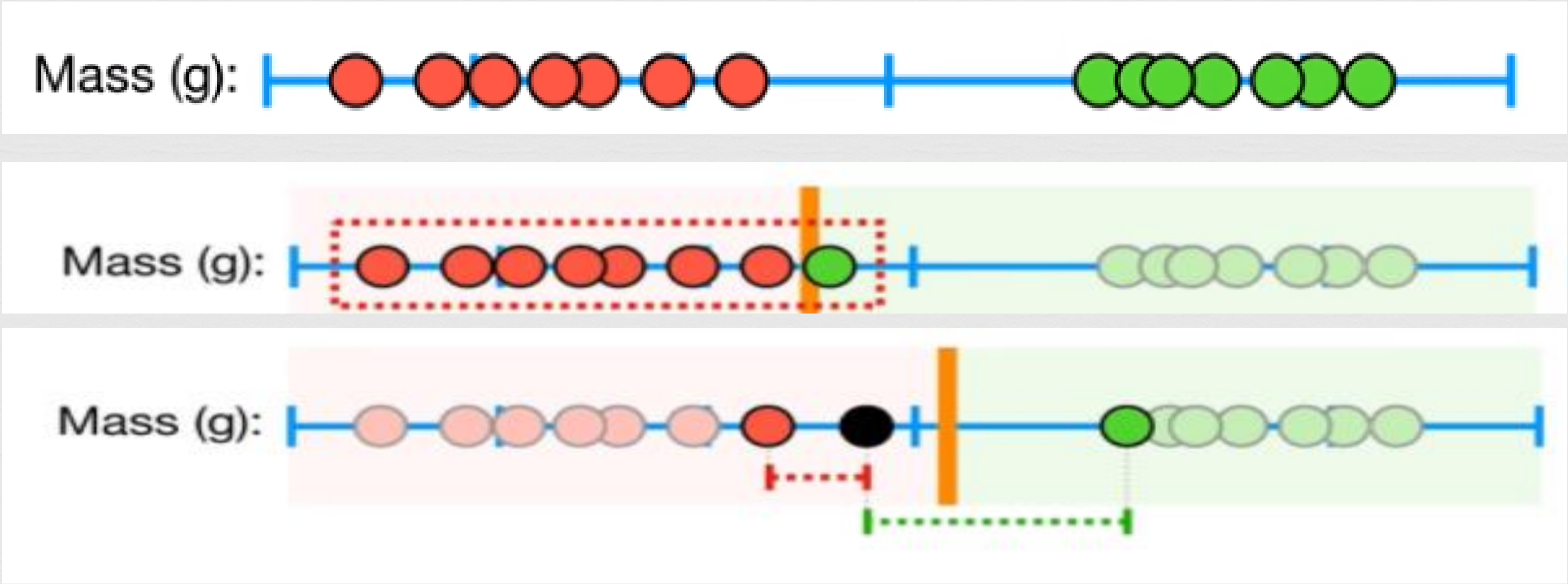
- Measurement of the **Mass of mice (g)**.
  - Red dots → not obese
  - Green dots → obese





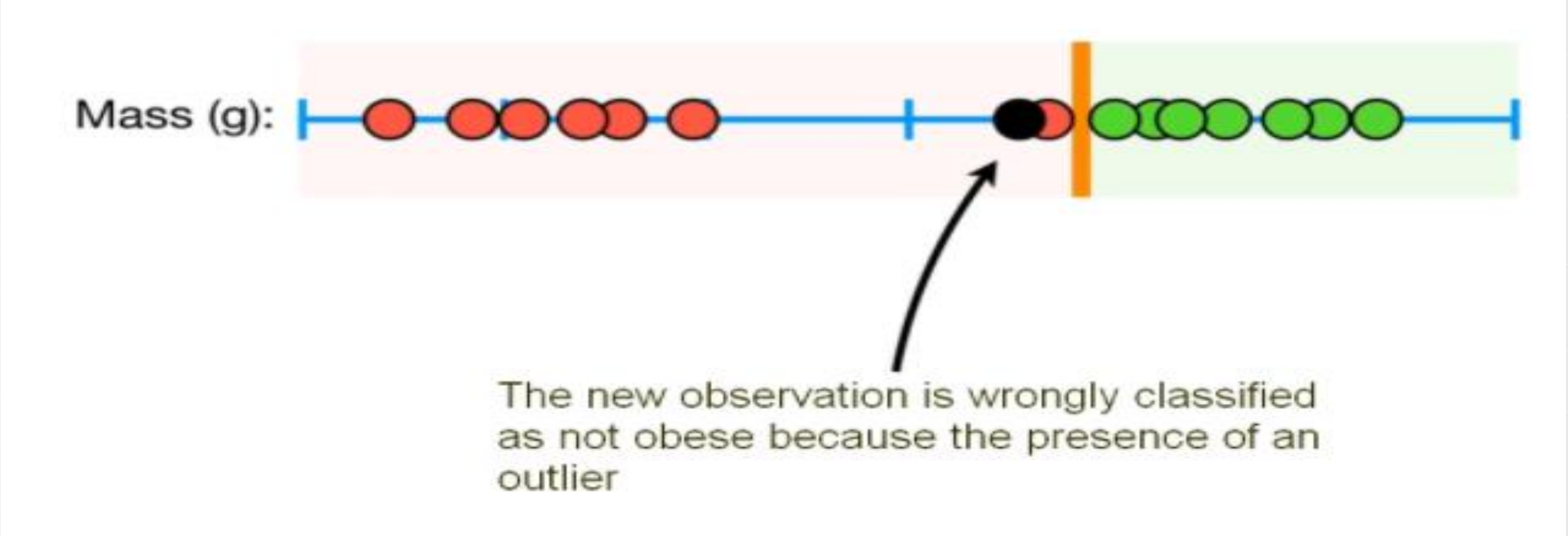
Support Vector Machine: Intuition

- Measurement of the **Mass of mice (g)**.
  - Red dots → not obese
  - Green dots → obese



Support Vector Machine: Intuition

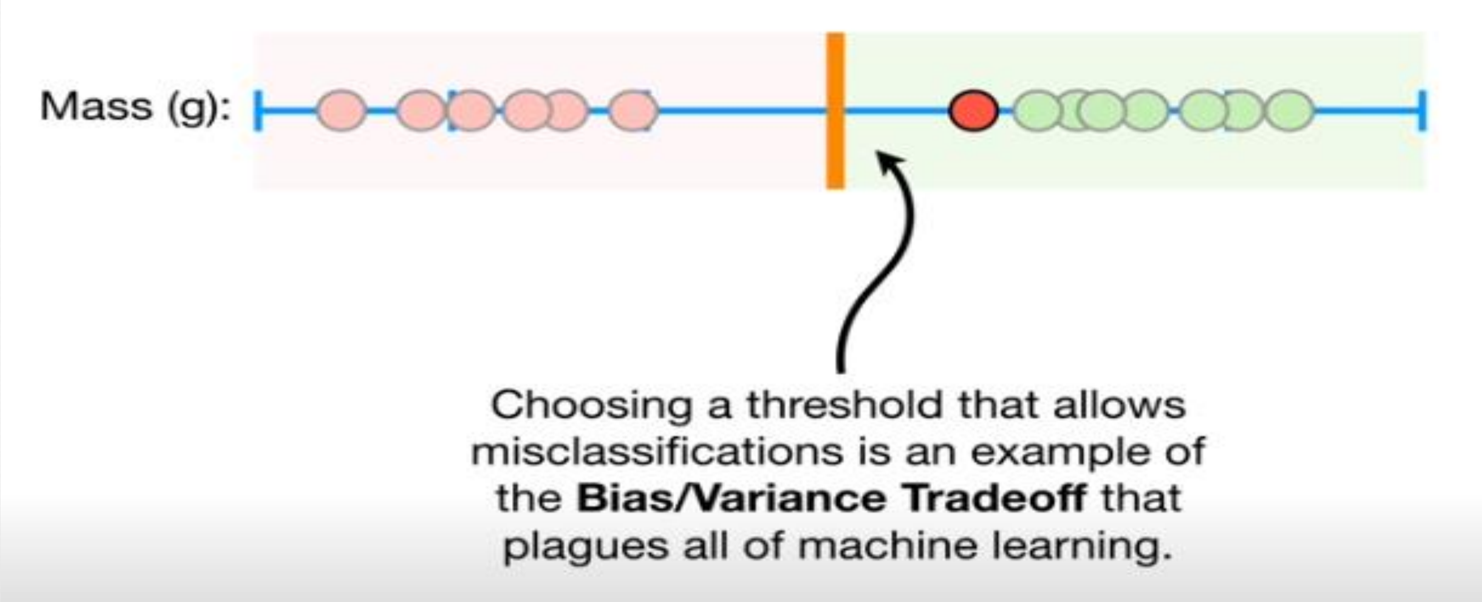
- Measurement of the **Mass of mice (g)**.
  - Red dots → not obese
  - Green dots → obese



Maximal Margin Classifier is super sensitive to **outliers**.

Support Vector Machine: Intuition

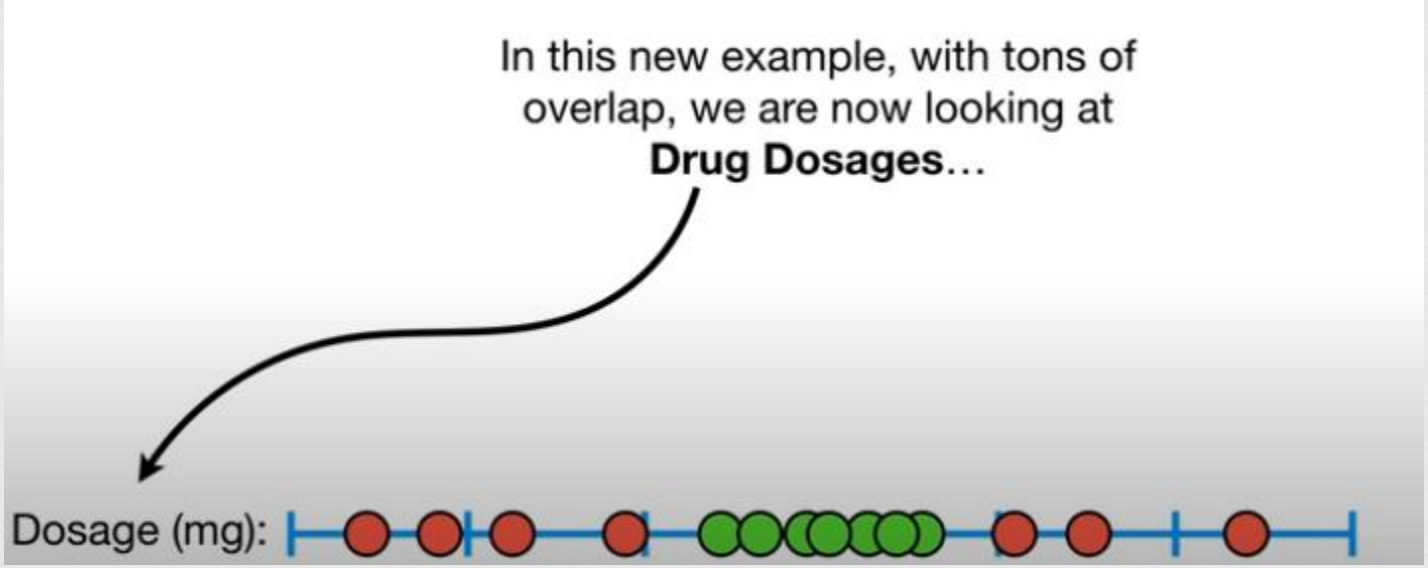
- Measurement of the **Mass of mice (g)**.
  - Red dots → not obese
  - Green dots → obese



Choosing a threshold that allows misclassifications.

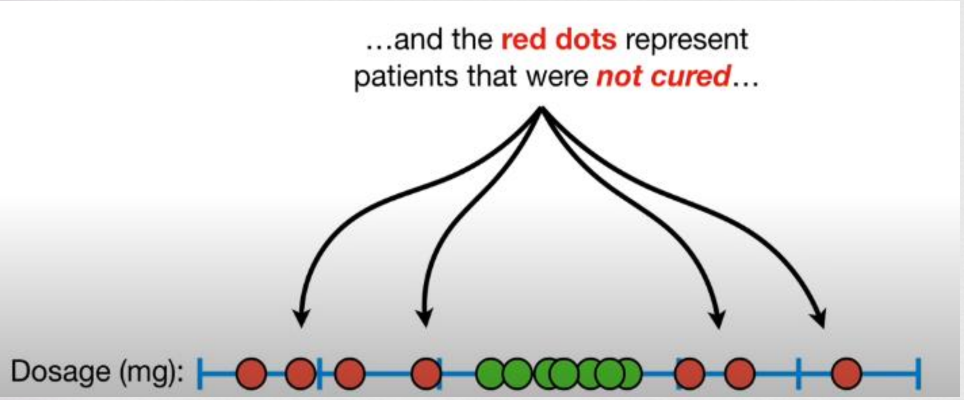
Support Vector Machine: Intuition

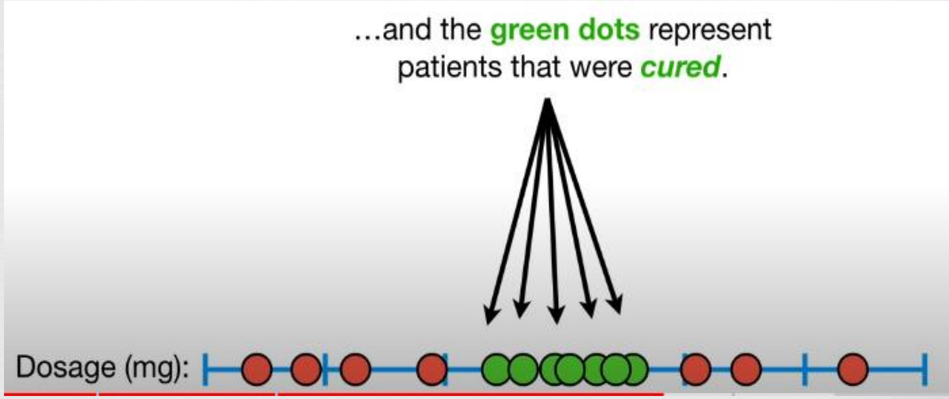
- Measurement of the **Drug Dosage**.
  - Red dots → patients were not cured
  - Green dots → patients were cured

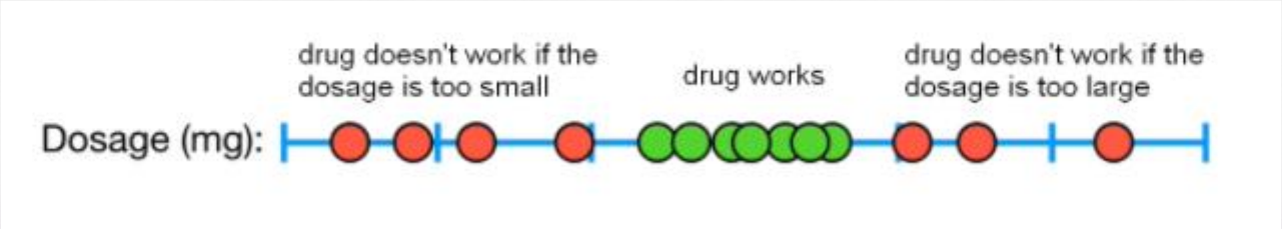


Support Vector Machine: Intuition

- Measurement of the **Drug Dosage**.
  - Red dots → patients were not cured
  - Green dots → patients were cured





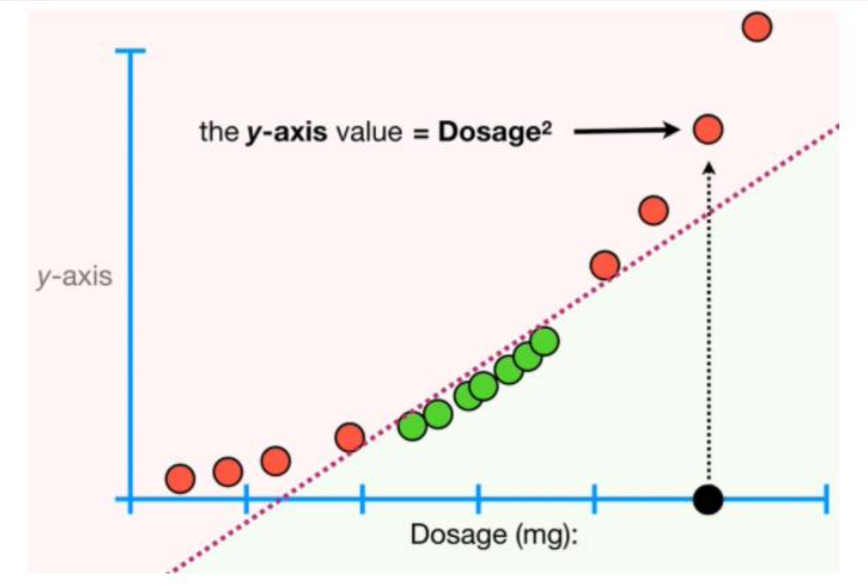


Support Vector Machine Intuition

- With the high overlapping depicted above, no matter where we put the classifier because will always make a lot of misclassifications.
- So, **Support Vector Classifiers** don’t perform well with this type of data.

Support Vector Machine Intuition

- Solution: We use the x-axis which represent the dosages we observed, but we also add an y-axis that will be the **square of the dosages**.



The main idea behind Support Vector Machines are:
1 - start with data in a relatively low dimension (in this example one dimension dosage in mg)
2 - move the data into a higher dimension (in this example from one to two dimensions)
3 - find a Support Vector Classifier that separates the higher dimensional data into two groups

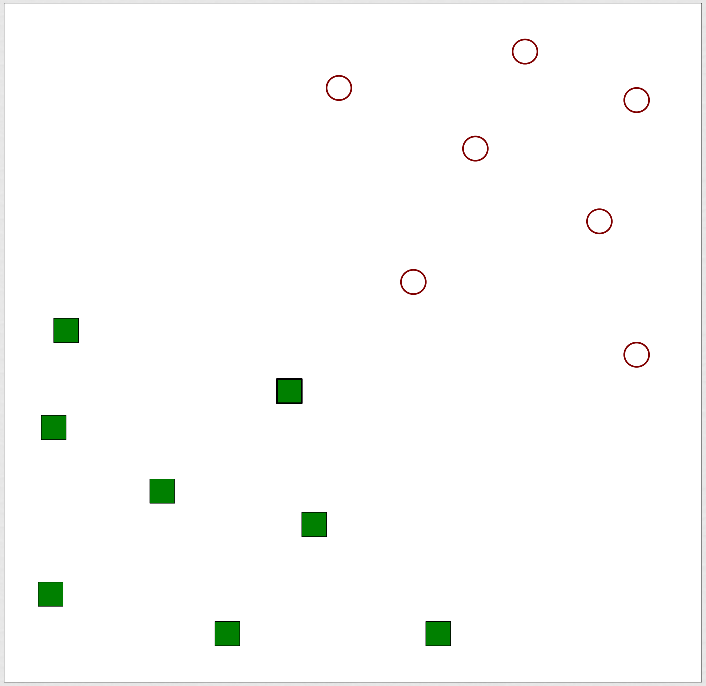
Support Vector Machine: Intuition

- When we use a Soft Margin to determine the location of a threshold, then we are using a **Soft Margin Classifier** aka a **Support Vector Classifier** to classify observations.
- The name Support Vector Classifier comes from the fact that the observations on the edge and within the Soft Margin are called **Support Vectors**.

Support Vector Machine

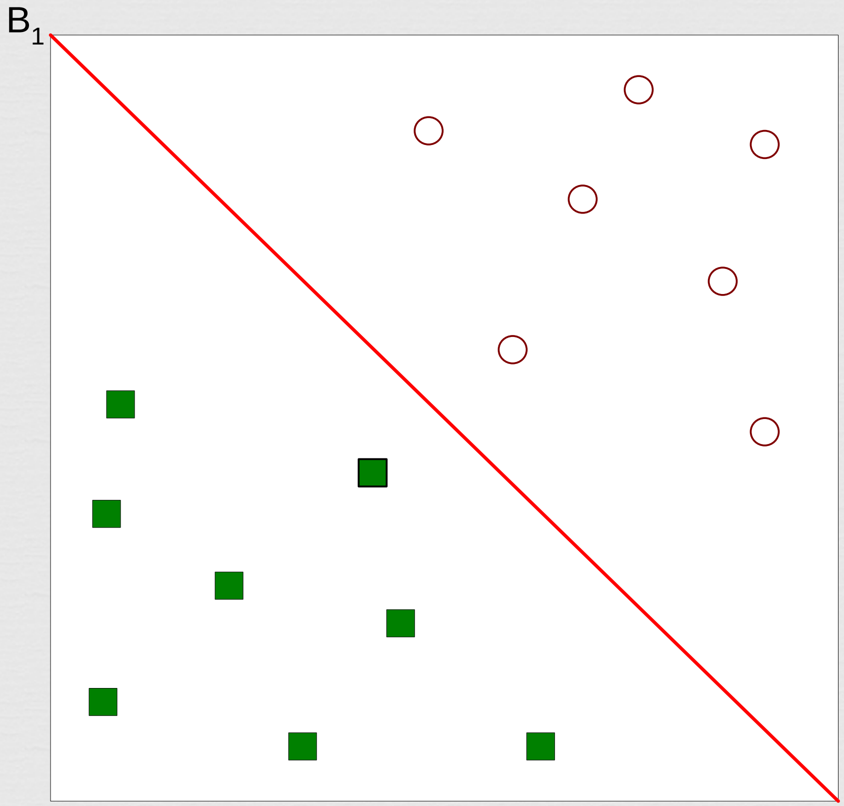
- SVM is a method for the classification of both linear and nonlinear data.
- SVM searches for the linear optimal separating hyperplane (i.e., a “decision boundary” separating the tuples of one class from another).
- Extend to patterns that are not linearly separable by transformations of original data to map into new space – the Kernel function.
- Support vectors are the data points that lie closest to the decision surface (or hyperplane).
  - They are the data points most difficult to classify
  - They have direct bearing on the optimum location of the decision surface

Support Vector Machines



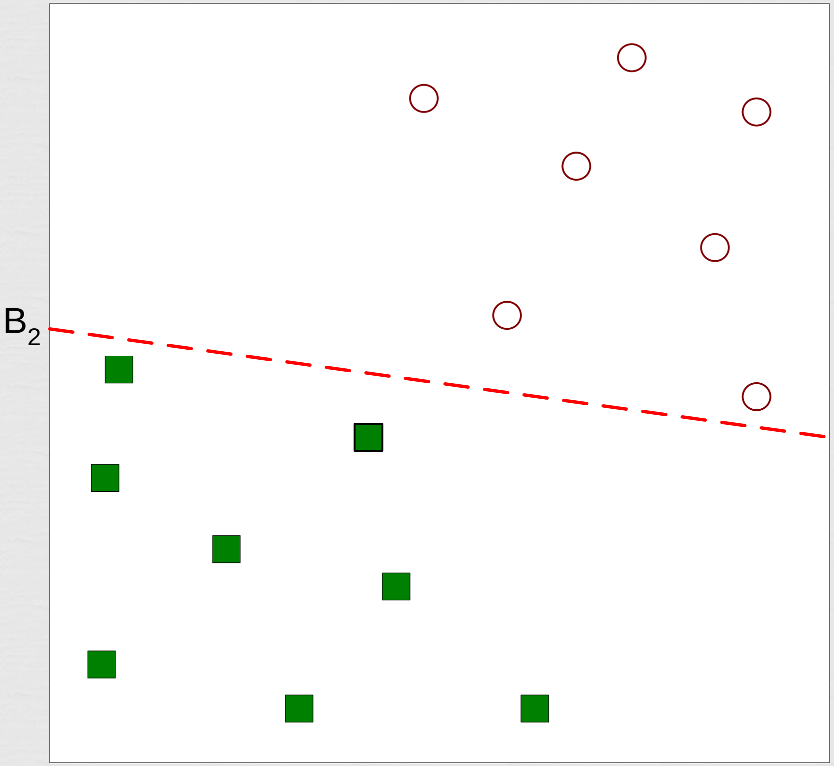
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



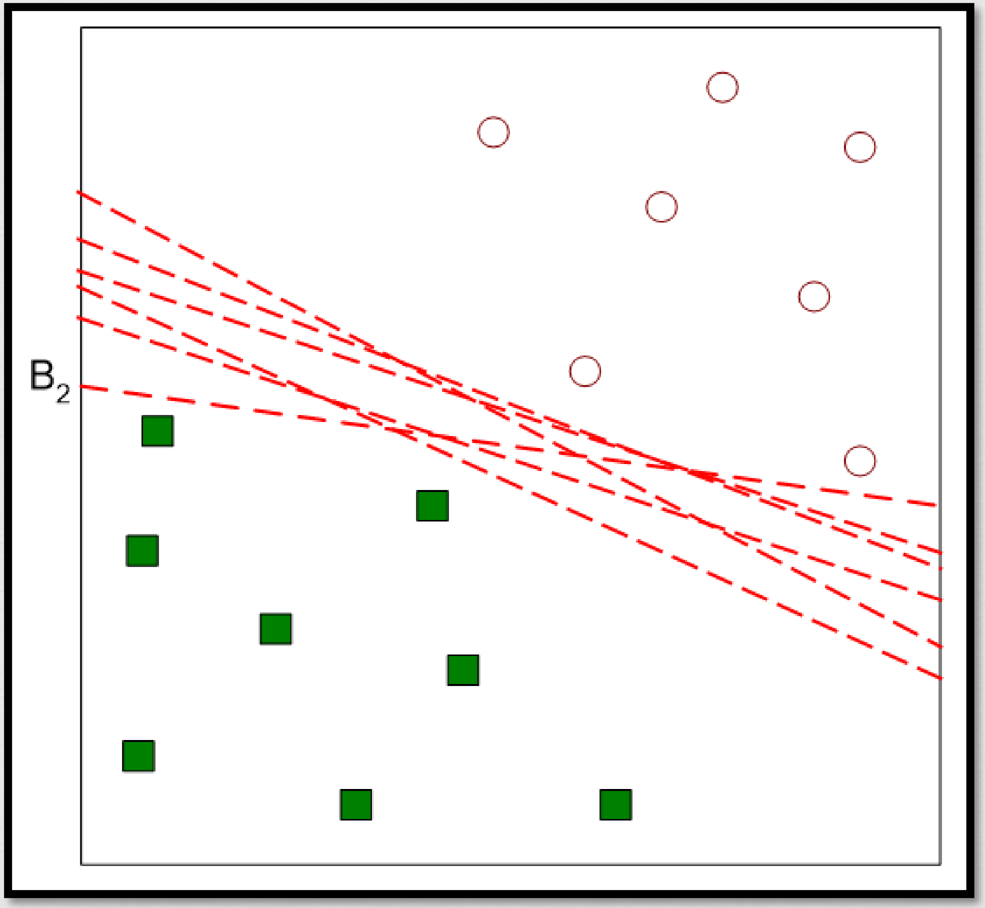
- One Possible Solution

Support Vector Machines



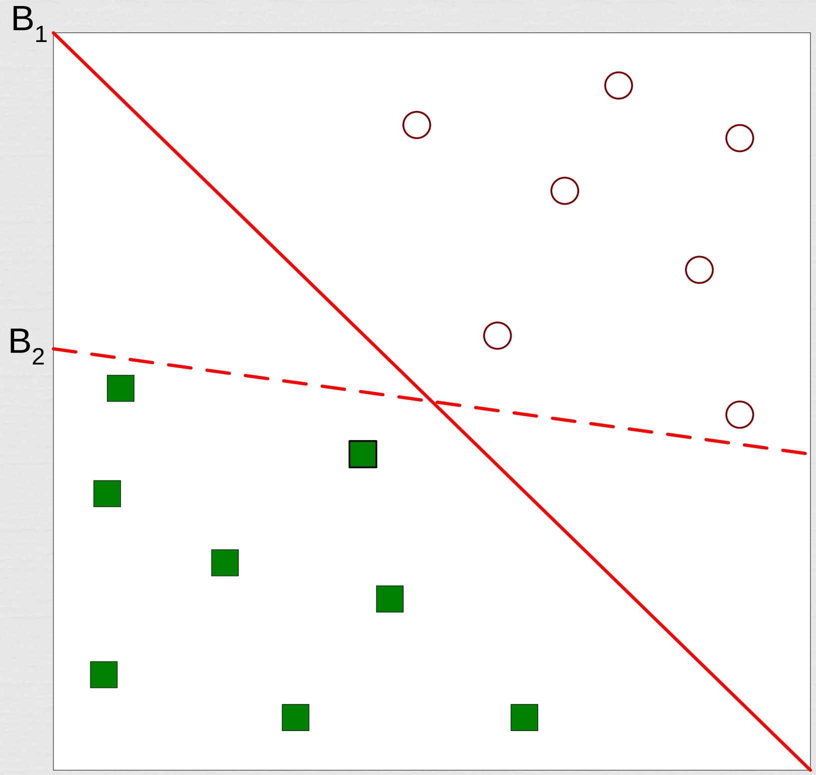
- Another possible solution

Support Vector Machines



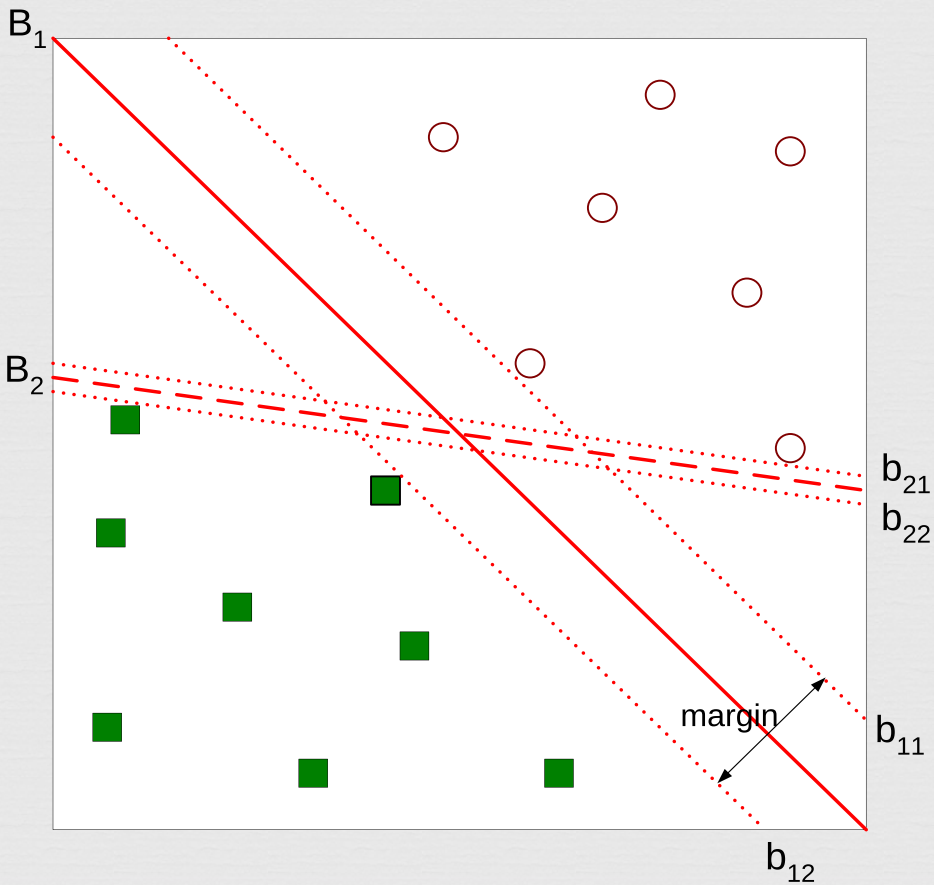
- Other possible solutions

Support Vector Machines



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

Support Vector Machines

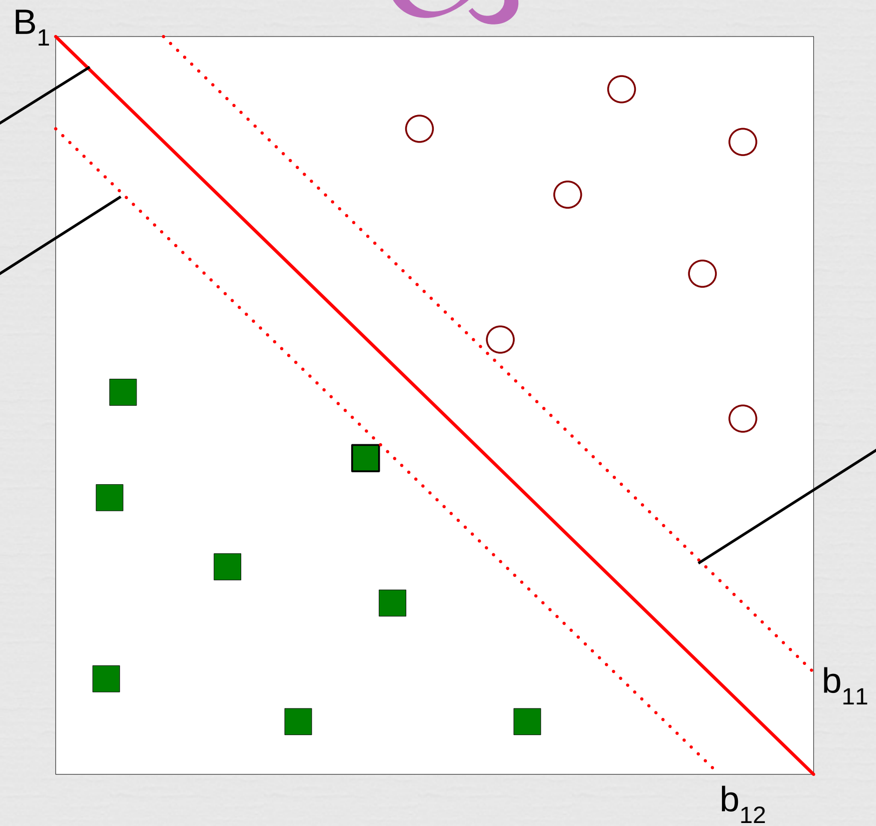


- Find hyperplane maximizes the margin => B1 is better than B2

Support Vector Machines

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

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



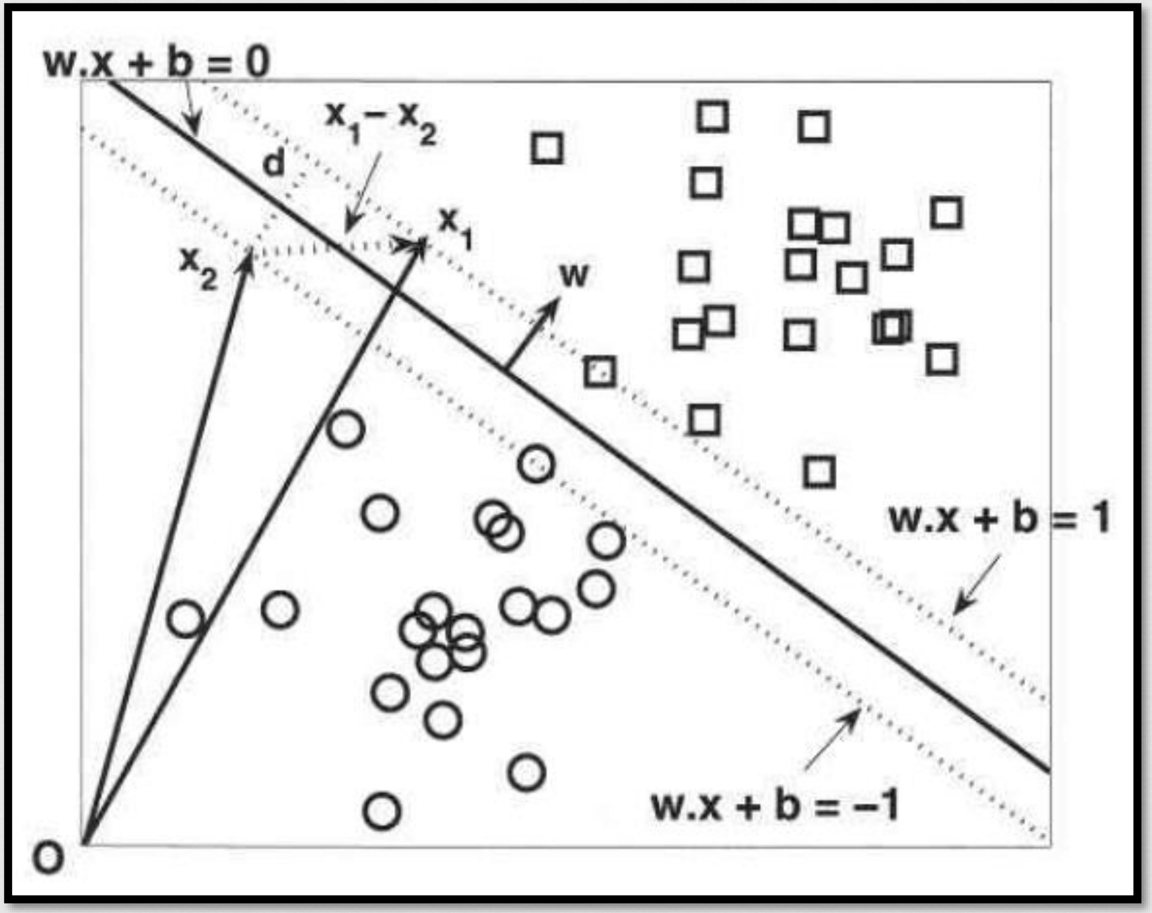
$$\vec{w} \bullet \vec{x} + b = +1$$

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

Rationale for Maximum Margin

- Decision boundaries with large margins tend to have better generalization capability than those with smaller margins.
- Intuitively, if the margin is small, then any slight perturbations to the decision boundary can have quite a significant impact on its classification.
- Classifiers that produce decision boundaries with small margins are more susceptible to model overfitting and tend to generalize poorly on previously unseen examples.

Maximum Margin Hyperplane



Linear SVM: Separable Case

- Binary classification problem
- N training Examples
- Each example is denoted by a tuple $(x_i, y_i)(i = 1,2, \ldots, N)$
- x_i is d-dimensional and $y_i \in \{-1,1\}$ such that y_i is +1 for positive example and y_i is -1 for negative example.
- Decision Boundary: $\boldsymbol{\vec{w}}.\boldsymbol{\vec{x}} + \boldsymbol{b} = \boldsymbol{0}$, where $\boldsymbol{\vec{w}}$ and $\boldsymbol{b}$ are parameters of the model.
- All training instances from class $\boldsymbol{y} = \boldsymbol{1}$ (i.e., the squares) must be located on or above the hyperplane $w.x + b = 1$, while those instances from class $\boldsymbol{y} = -\boldsymbol{1}$ (i.e., the circles) must be located on or below the hyperplane $\boldsymbol{w}.\boldsymbol{x} + \boldsymbol{b} = -\boldsymbol{1}$.

Linear SVM: Separable Case

- If x_+ and x_- are any two points located above the positive and negative marginal boundaries, respectively, then

$$\vec{\boldsymbol{w}}.x_+ + \boldsymbol{b} \geq \boldsymbol{+1} \text{ and } \vec{\boldsymbol{w}}.x_- + \boldsymbol{b} \leq \boldsymbol{-1}$$

- Since, y_i is +1 for positive example and y_i is -1 for negative example. The compact form of the above two constraints is a follows:

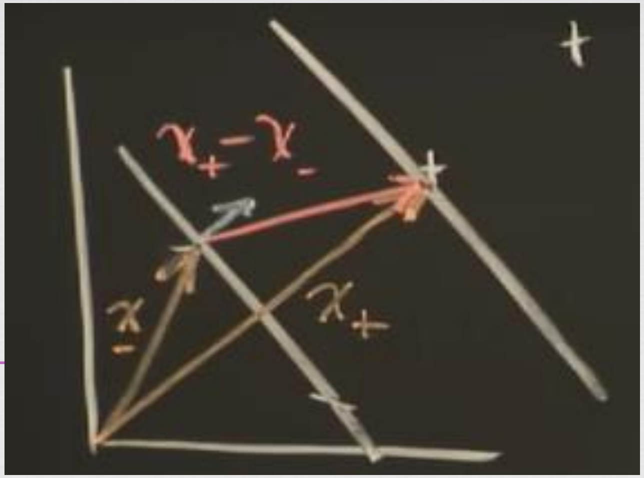
$$y_i(\vec{\boldsymbol{w}}.x + \boldsymbol{b}) \geq \boldsymbol{+1}$$

Learning a linear SVM Model

- Training Phase: Estimation of $\boldsymbol{w}$ and $\boldsymbol{b}$ of the decision boundary from training data.
- The Parameters are chosen such the following condition is met.

$$\boldsymbol{y_i(w.x_i + b) \geq 1, i = 1, 2, \ldots, N}.$$

Margin of a Linear Classifier



- Consider a square located on the positive marginal hyperplane => $\boldsymbol{b_{i1}}: \vec{\boldsymbol{w}}. x_+ + \boldsymbol{b} = \boldsymbol{1}$
- Consider a circle located on the negative marginal hyperplane => $\boldsymbol{b_{i2}}: \vec{\boldsymbol{w}}. x_- + \boldsymbol{b} = \boldsymbol{-1}$
- The margin of decision boundary is the distance between these two hyperplanes.
- Since x_+ is a point located on $\boldsymbol{b_{i1}}$ and x_- is a point located on $\boldsymbol{b_{i2}}$.
- $x_+ - x_-$ is a vector directed from x_- to x_+.
- Direction of $\vec{\boldsymbol{w}}$ is perpendicular to the decision boundary, therefore, normalizing $\vec{\boldsymbol{w}}$ will yield a unit vector in the perpendicular direction.
- Therefore, the margin (distance between $\boldsymbol{b_{i1}}$ and $\boldsymbol{b_{i2}}$) can be computed as $\frac{\vec{\boldsymbol{w}}}{\|\boldsymbol{w}\|}.(x_+ - x_-)$ (Since $\vec{\boldsymbol{b}}.\vec{\boldsymbol{a}} = \|\boldsymbol{b}\|\|\boldsymbol{a}\|\boldsymbol{cos\theta}$ => $\frac{\vec{\boldsymbol{b}}}{\|\boldsymbol{b}\|}.\vec{\boldsymbol{a}} = \|\boldsymbol{a}\|\boldsymbol{cos\theta}$)

$$= \frac{\vec{\boldsymbol{w}}.x_+}{\|\boldsymbol{w}\|} - \frac{\vec{\boldsymbol{w}}.x_-}{\|\boldsymbol{w}\|} = \frac{\boldsymbol{1-b-(-1-b)}}{\|\boldsymbol{w}\|} = \frac{\boldsymbol{2}}{\|\boldsymbol{w}\|}$$

Support Vector Machines

- We want to maximize: $Margin = \frac{2}{\|w\|}$

  - Which is equivalent to minimizing: $L(w) = \frac{\|w\|}{2}$

  - But subjected to the following constraints:

$$\boldsymbol{y_i(w.x_i + b) \geq 1, i = 1, 2, \dots, N}$$

    - This is a constrained optimization problem
      - Numerical approaches to solve it (e.g., quadratic programming)

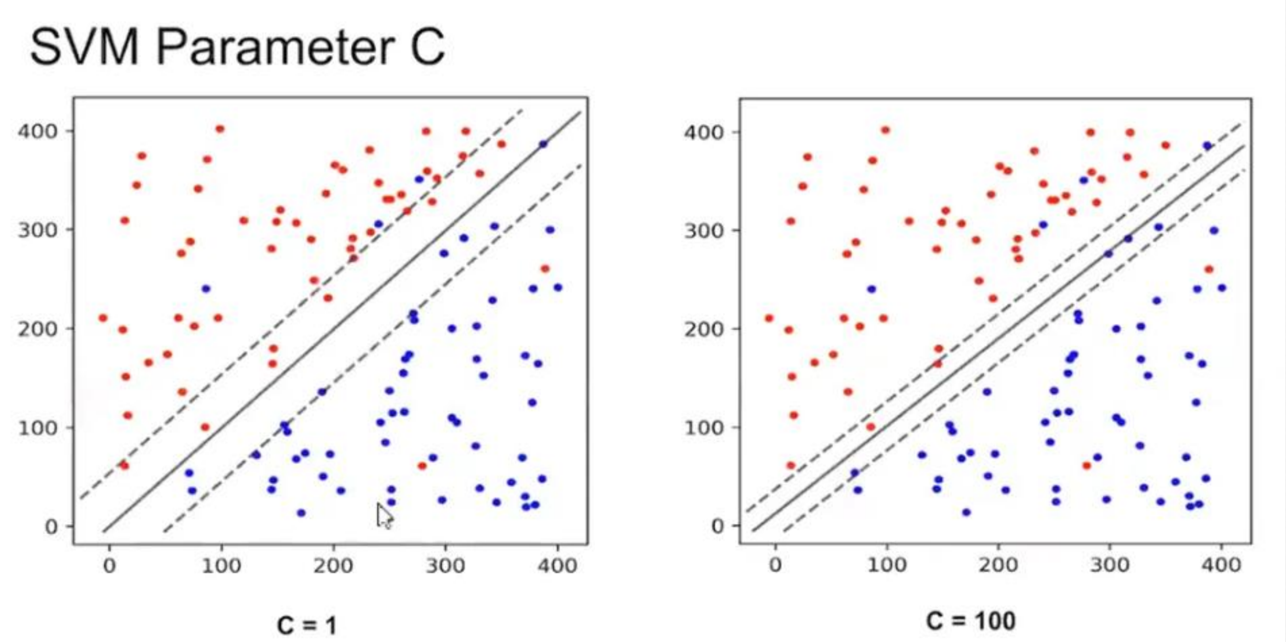
Hyperparameters of linear SVM

- Objective (Cost) Function:

$$\underset{\vec{w},b}{\text{min}imize}\,\frac{\|w\|}{2}+C\sum_{i=1}^{n}\xi_i$$

- C is the hyperparameter that controls the number of misclassification points.
- Eta (greek symbol) is the summation of misclassified points from marginal plane.

Hyperparameters of linear SVM



1.Smaller C: When C is small => SVM places a higher priority on achieving a wide margin, even if that means allowing more misclassifications. In this case, the SVM is more tolerant of misclassified points and focuses on finding a larger margin.

2.Larger C: When C is large, the SVM becomes more sensitive to misclassifications and tries to minimize them as much as possible. This can lead to a narrower margin to correctly classify more points.