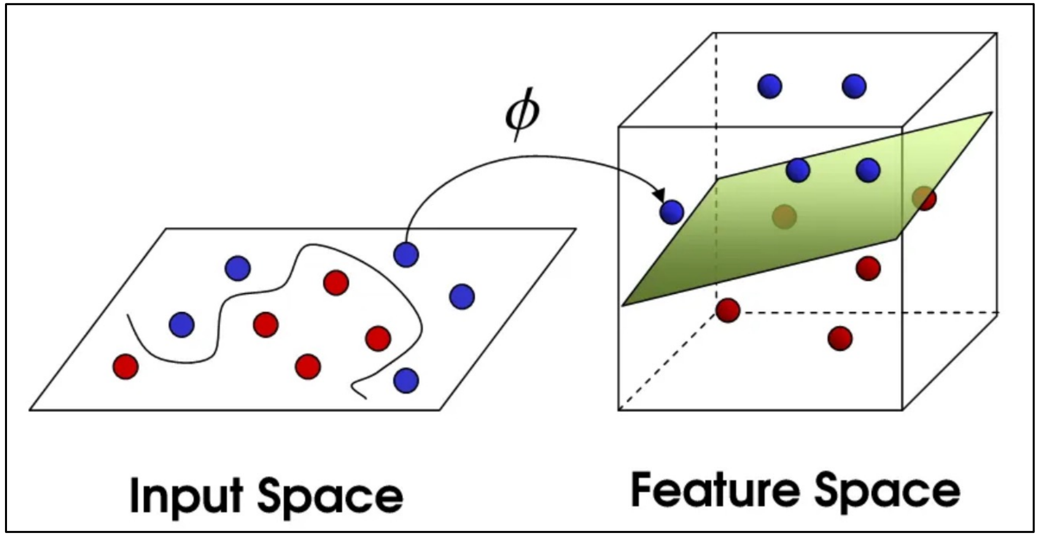
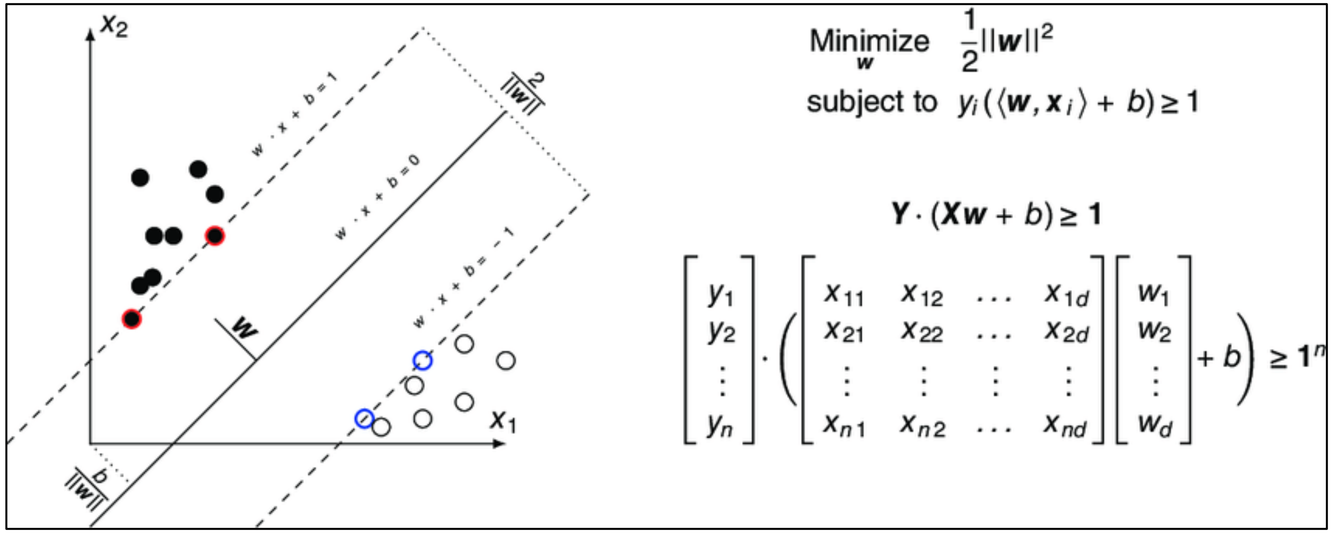
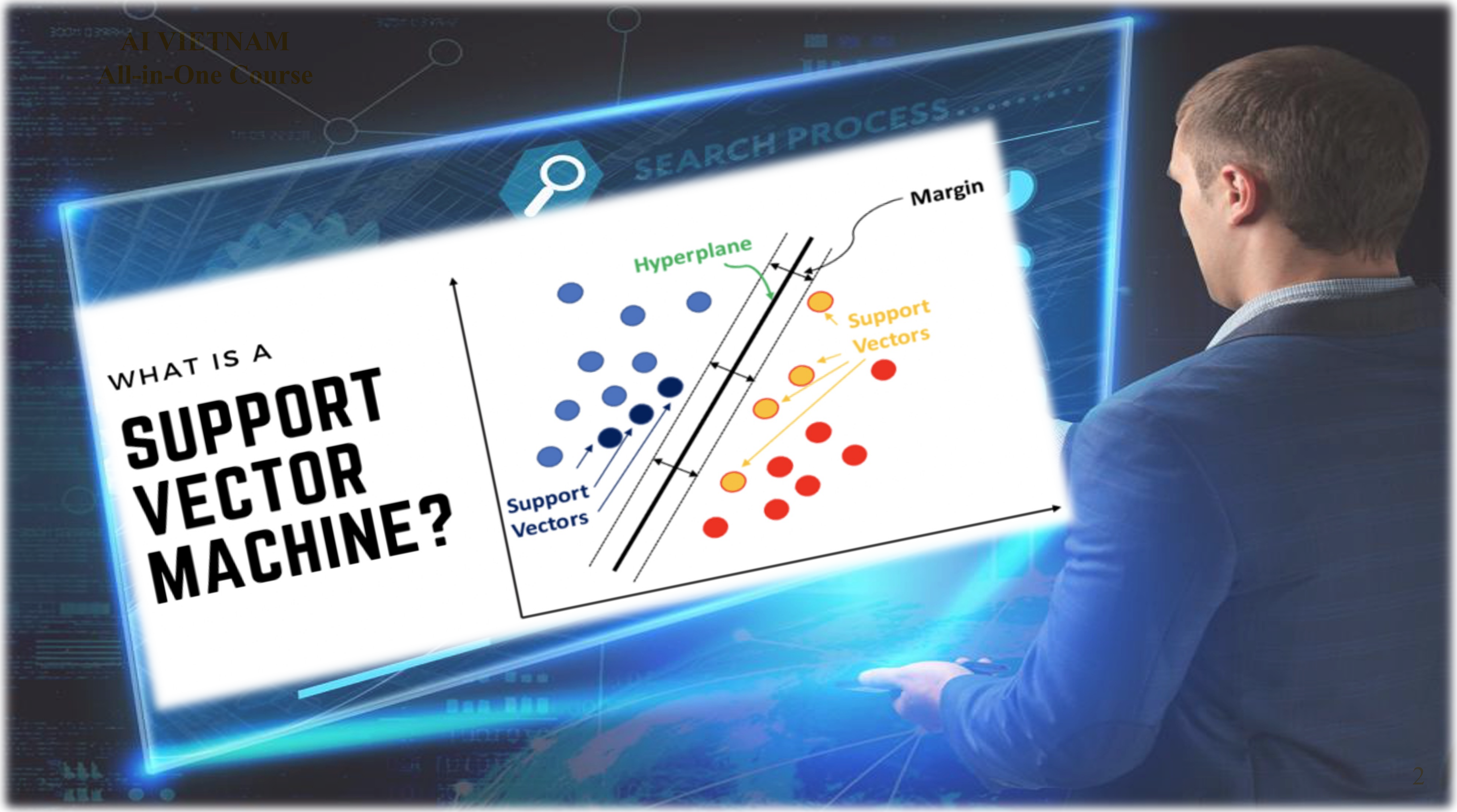
Support Vector Machine (Another View)





Vinh Dinh Nguyen
PhD in Computer Science

Year 2023


SEARCH PROCESS
WHAT IS A
SUPPORT
VECTOR
MACHINE?
Margin
Hyperplane
Support
Vectors
Support
Vectors

Outline

- Linear Regression to Logistic Regression
- Logistic Regression to Support Vector Machine
- Example

Outline

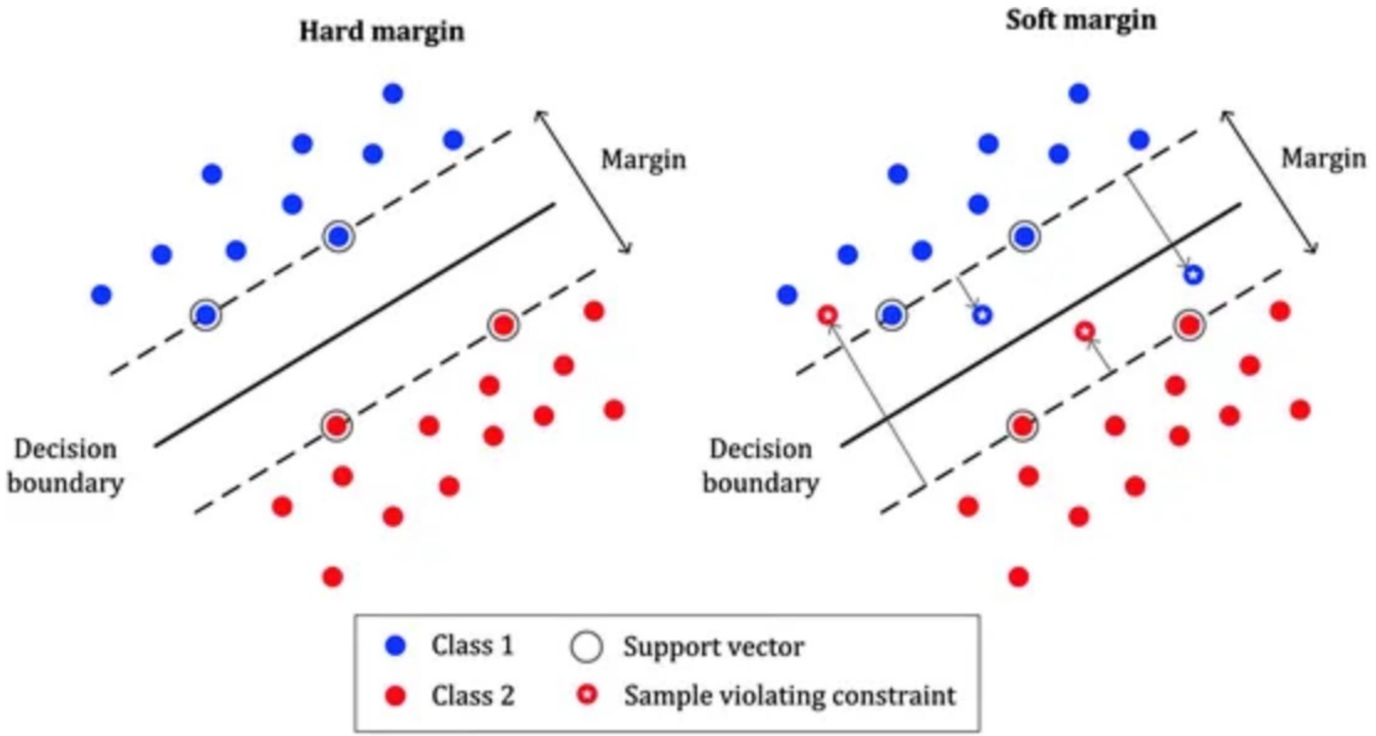
- Linear Regression to Logistic Regression
- Logistic Regression to Support Vector Machine
- Example

Review – Linear SVM

Primal Problem

$$\min \frac{1}{2} ||W||_2^2 + C \sum_i^N \varepsilon_i \text{ such that } Y * (w^T X + b) \geq 1 - \varepsilon_i$$

$\varepsilon_i \geq 0$



Review- Linear SVM

Primal Problem

$$\min \frac{1}{2} \|W\|_2^2 + C \sum_i^N \varepsilon_i \text{ such that } Y * (w^T X + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

Another form:

$$\text{Objective Function}: \min_{\beta, b, \xi_i} \left\{ \frac{\|\beta^2\|}{2} + C \sum_{i=1}^{n} (\xi_i)^k \right\}$$

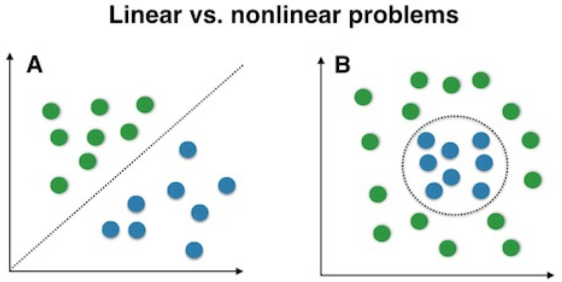
$$\text{s.t Linear Constraint}: y_i(\beta^T x_i + b) \geq 1 - \xi_i, \text{ where } \xi_i \geq 0$$

C and k are constants which balances the cost of misclassification. The $\sum_{i=1}^{n} (\xi_i)^k$ is the loss term and C is a HyperParameter which controls the tread-off between maximizing the margin and minimizing the loss.

The HyperParameter C is also called as *Regularization Constant.*

If k = 1 , then the loss is named as Hinge Loss and if k =2 then its called Quadratic Loss

How bout Non-Linear Case?



Review- Linear SVM

Primal Problem

$$\min \frac{1}{2} ||W||_2^2 + C \sum_i^N \varepsilon_i \text{ such that } Y * (w^T X + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0$$

Another form:

$$\text{Objective Function}: \min_{\beta, b, \xi_i} \left\{ \frac{||\beta^2||}{2} + C \sum_{i=1}^{n} (\xi_i)^k \right\}$$

$$\text{s.t Linear Constraint}: y_i(\beta^T x_i + b) \geq 1 - \xi_i, \text{ where } \xi_i \geq 0$$

`C` and `k` are constants which balances the cost of misclassification. The $\sum_{i=1}^{n} (\xi_i)^k$ is the loss term and C is a HyperParameter which controls the tread-off between maximizing the margin and minimizing the loss.

The HyperParameter C is also called as *Regularization Constant.*

If k = 1 , then the loss is named as Hinge Loss and if k =2 then its called Quadratic Loss

However, this way we won't be able use the objective function to solve for non-linear cases. Hence, we will find an equivalent problem named **Dual Problem** and solve that using **Lagrange Multipliers.**

Dualily

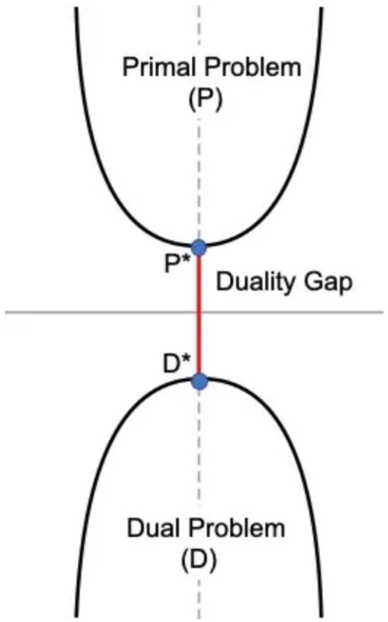
Duality

" *In mathematical optimization theory, **duality** means that optimization problems may be viewed from either of two perspectives, the primal problem or the dual problem (**the duality principle**). The solution to the dual problem provides a lower bound to the solution of the primal (minimization) problem.*

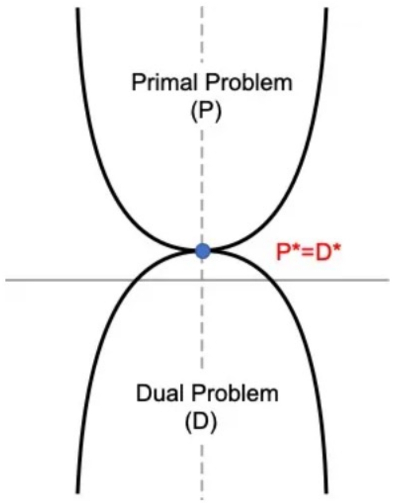
Wikipedia

Dual Problem (Bài toán đối ngẫu_

- Primal Problem is something that we want to minimize. In the diagram, **P*** minimizes the Primal Objective **P**.
- Dual Problem is something we want to maximize. Here we want to convert the Primal Problem (**P**) to a Dual Problem (**D**). **D*** maximizes the Dual Objective **D**.
- Sometimes solving the Dual Problem is same as solving the Primal Problem.
- $(P^* - D^*)$ is called as the **Duality Gap**.
- If $P^* - D^* > 0$ we can say **weak duality holds**.



- The goal will be to find in which condition we can have $P^* - D^* = 0$ and we can say that **strong duality holds**.
- In this picture, we can see that maximizing the Dual Problem is same as Minimizing the Primal Problem.
- There are some conditions named **KKT condition**, needs to hold in order to have $P^* - D^* = 0$
- Next, we will talk about **Lagrange Multiplier**, which will help us to determine when we can find strong duality.



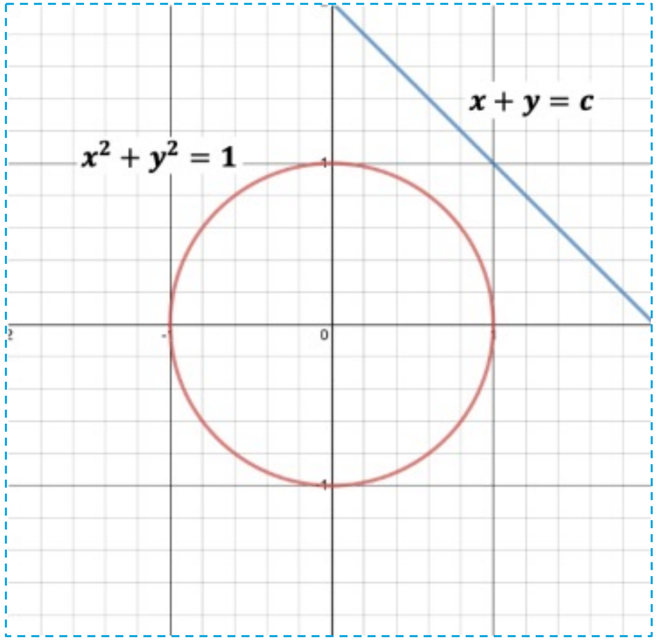
Warm-up

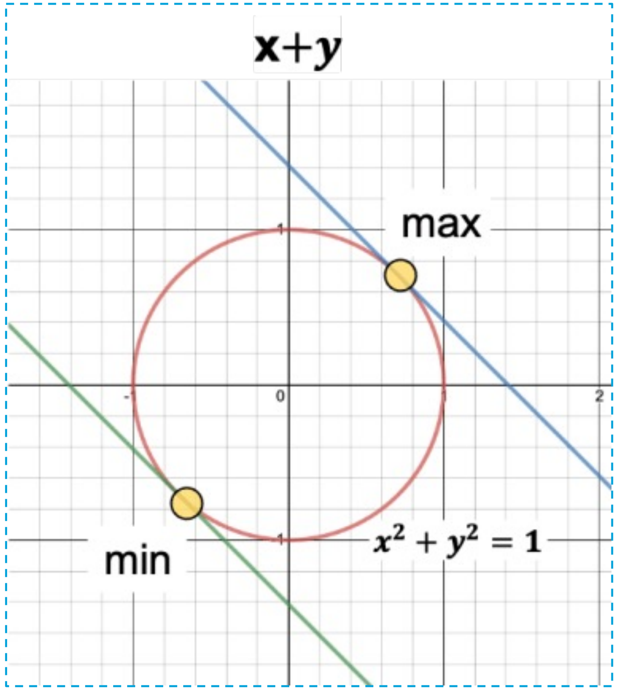
Primal Problem is helpful in solving Linear SVM using SGD. Remember we need to optimize D+1 (where D is the dimension) parameters in Primal Problem.

Primal Problem

$$\max_{x,y} x + y$$
$$\text{s.t. } x^2 + y^2 = 1$$

- Here is the plot of the constraint $x^2 + y^2 = 1$ in red, which is a circle.
- $x + y$ is basically the entire plane.
- The blue line is an arbitrary representation of $x + y = c$





Lagrange Multiplier

Primal Problem

$$\max_{x,y} x + y$$
$$\text{s.t. } x^2 + y^2 = 1$$

$$\max_{x,y} f(x, y)$$
$$\text{such that } g(x, y) = c$$

$$L(x, y, \alpha) = f(x, y) - \alpha\Big(g(x, y) - c\Big)$$
$$= x + y - \alpha(x^2 + y^2 - 1)$$

$$\delta_x L(x, y, \alpha) = 1 - 2\alpha x = 0$$
$$x = \frac{1}{2\alpha}$$
$$\delta_y L(x, y, \alpha) = 1 - 2\alpha y = 0$$
$$y = \frac{1}{2\alpha}$$
$$\delta_\alpha L(x, y, \alpha) = x^2 + y^2 - 1 = 0$$
$$\frac{1}{2\alpha}^2 + \frac{1}{2\alpha}^2 - 1 = 0$$
$$\frac{1}{2\alpha^2} = 1$$
$$\alpha = \pm\frac{1}{\sqrt{2}}$$

Since we are tying to maximize $x + y$, we will consider only the positive values.

Final Result:

$$x = \frac{1}{\sqrt{2}}$$
$$y = \frac{1}{\sqrt{2}}$$

$$f(x, y) = x + y$$
$$= \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}$$
$$= \sqrt{2}$$

$$\max_x f(x)$$
$$\text{s.t. } g(x) = 0$$

Assume x has D dimensions, then there will be total D+1 unknowns for the following function,

If you have n constraints then there will be total D+n unknowns.

$$\max_x f(x)$$
$$\text{s.t. } g_1(x) = 0, g_2(x) = 0, \ldots, g_n(x) = 0$$

We can then define the **Lagrangian** as following,

$$L(x_1, \ldots, x_D, \alpha_1, \ldots, \alpha_n) = f(x) - \sum_{i=1}^{n} \alpha_i g_i(x)$$

However, in order to find the solution for the variables, <u>it wont be enough only to take the gradients and set those to zero due to the Inequality Constraints.</u>

Inequality Constraint

$$\max_x f(x)$$
$$\text{s.t. } g_i(x) \leq 0 \text{ , } \forall i = 1..n$$
$$h_i(x) = 0 \text{ , } \forall j = 1..m$$

We can define the **Lagrangian** as (One constant for each constraint),

$$L(x, \alpha, \lambda) = f(x) + \sum_{i=1}^{n} \alpha_i g_i(x) + \sum_{j=1}^{m} \lambda_j h_j(x)$$

Setting $\Delta_{x,\lambda} L = 0$ still gives two of the conditions, but for the Inequality Constraint, we need to have 3 additional conditions. Hence instead of total 3, we will now have total 5 conditions.

$$\alpha_i g_i(x) = 0 \text{ , } \forall i = 1..n$$
$$g_i(x) \leq 0 \text{ , } \forall i = 1..n$$
$$\alpha_i \geq 0 \text{ , } \forall i = 1..n$$
$$\delta_{x_d} L = 0 \text{ , } \forall d = 1..D$$
$$\delta_{\lambda_j} L = 0 \text{ , } \forall j = 1..m$$

These above five conditions are called as KKT (Karush–Kuhn–Tucker) conditions and they must be met for strong duality, i.e for P∗−D∗=0 to be true.

Duality: Hard Margin Classifier

Change *Hard Margin Classifier's Objective Function* from *Primal Problem* to *Dual Problem* using Lagrange Multiplier and KKT Conditions.

$$\text{Objective Function}: \min_{\beta,b}\left\{\frac{||\beta^2||}{2}\right\}$$

$$\text{s.t Linear Constraint}: y_i(\beta^T x_i + b) \geq 1 \text{ , } \forall x_i \in D$$

The constraint can be redefined as following (This is required for the 2^{nd} KKT Condition $g_i(x) \leq 0$),

$$\text{s.t Linear Constraint}: 1 - y_i(\beta^T x_i + b) \leq 0 \text{ , } \forall x_i \in D$$

The **Lagrangian** can defined as following,

$$\begin{aligned} \min L &= \frac{||\beta^2||}{2} + \sum_{i=1}^{n} \alpha_i(1 - y_i(\beta^T x_i + b)) \\ &= \frac{||\beta^2||}{2} - \sum_{i=1}^{n} \alpha_i(y_i(\beta^T x_i + b) - 1) \end{aligned}$$

KKT (Karush–Kuhn–Tucker) conditions

$$\begin{aligned} \alpha_i g_i(x) &= 0 \text{ , } \forall i = 1..n \\ g_i(x) &\leq 0 \text{ , } \forall i = 1..n \\ \alpha_i &\geq 0 \text{ , } \forall i = 1..n \\ \delta_{x_d} L &= 0 \text{ , } \forall d = 1..D \\ \delta_{\lambda_j} L &= 0 \text{ , } \forall j = 1..m \end{aligned}$$

$$\delta_\beta L = \beta - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\beta = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\text{and } \delta_b L = \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i(1 - y_i(\beta^T x_i + b)) = 0$$
$$\text{and } \alpha_i \geq 0$$

$$1 - y_i(\beta^T x_i + b) \leq 0 \text{ , } \forall x_i \in D$$

Duality: Hard Margin Classifier

Change Hard Margin Classifier's Objective Function from Primal Problem to Dual Problem using Lagrange Multiplier and KKT Conditions.

The **Lagrangian** can defined as following,

$$\min L = \frac{||\beta^2||}{2} + \sum_{i=1}^{n} \alpha_i (1 - y_i(\beta^T x_i + b))$$
$$= \frac{||\beta^2||}{2} - \sum_{i=1}^{n} \alpha_i (y_i(\beta^T x_i + b) - 1)$$

$$\delta_\beta L = \beta - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$
$$\beta = \sum_{i=1}^{n} \alpha_i y_i x_i$$
$$\text{and } \delta_b L = \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\alpha_i (1 - y_i(\beta^T x_i + b)) = 0 \text{ and } \alpha_i \geq 0$$

$$1 - y_i(\beta^T x_i + b) \leq 0 \,,\, \forall x_i \in D$$

Dual Lagrangian Objective Function

$$L_{dual} = \frac{||\beta^2||}{2} - \sum_{i=1}^{n} \alpha_i (y_i(\beta^T x_i + b) - 1)$$
$$= \frac{1}{2}\beta^T \beta - \sum_{i=1}^{n} \alpha_i y_i \beta^T x_i - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i$$
$$= \frac{1}{2}\beta^T \beta - \beta^T \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right) - b\left(\sum_{i=1}^{n} \alpha_i y_i\right) + \sum_{i=1}^{n} \alpha_i$$
$$= \frac{1}{2}\beta^T \beta - \beta^T (\beta) - b(0) + \sum_{i=1}^{n} \alpha_i$$
$$= -\frac{1}{2}\beta^T \beta + \sum_{i=1}^{n} \alpha_i$$
$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Duality: Hard Margin Classifier

$$
\begin{aligned}
L_{dual} &= \frac{||\beta^2||}{2} - \sum_{i=1}^{n} \alpha_i (y_i(\beta^T x_i + b) - 1) \\
&= \frac{1}{2}\beta^T\beta - \sum_{i=1}^{n} \alpha_i y_i \beta^T x_i - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i \\
&= \frac{1}{2}\beta^T\beta - \beta^T\Big(\sum_{i=1}^{n} \alpha_i y_i x_i\Big) - b\Big(\sum_{i=1}^{n} \alpha_i y_i\Big) + \sum_{i=1}^{n} \alpha_i \\
&= \frac{1}{2}\beta^T\beta - \beta^T\Big(\beta\Big) - b\Big(0\Big) + \sum_{i=1}^{n} \alpha_i \\
&= -\frac{1}{2}\beta^T\beta + \sum_{i=1}^{n} \alpha_i \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\end{aligned}
$$

L should be minimized w.r.t β and b, and should be maximized w.r.t α_i. Hence, instead of **minimizing** the **Primal Problem**, we can now **maximize** the **Dual Problem** (**P*=D*** as the KKT conditions are now satisfied). So we can write the following,

$$\textbf{Objective Function:}\ \max_{\alpha} L_{dual} = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{x_i^T x_j}$$

$$\textbf{Linear Constraints:}\ \alpha_i \geq 0, \forall i \in D, \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Once we have got the α_i values, we can calculate the weight vector β and bias **b**. As per the KKT conditions we had,

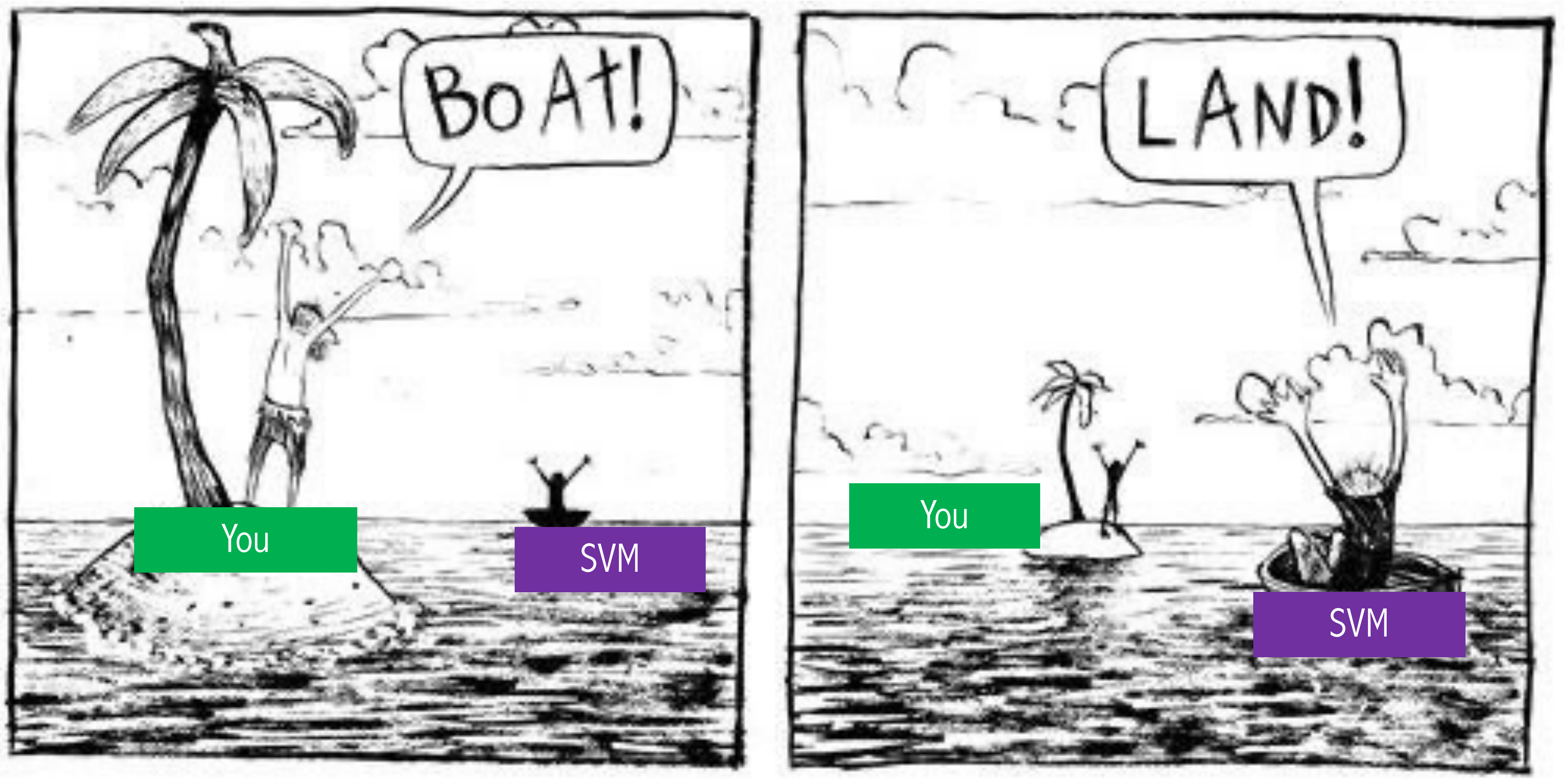
$$\boxed{\beta = \sum_{i=1}^{n} \alpha_i y_i x_i}$$

$$\alpha_i (1 - y_i(\beta^T x_i + b)) = 0$$

We can now find the optimal Hyperplane given β and **b**. For any new point **z**, we can predict the class using following,

$$\hat{y} = \text{sign}(\beta^T z + b)$$

Another View of the SVM



Linear Regression

Develop a program to predict the price of the house based on the area of the house. Can we use Linear Regression to solve this problem?

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$h_\theta(x) = [-\infty, +\infty]$$

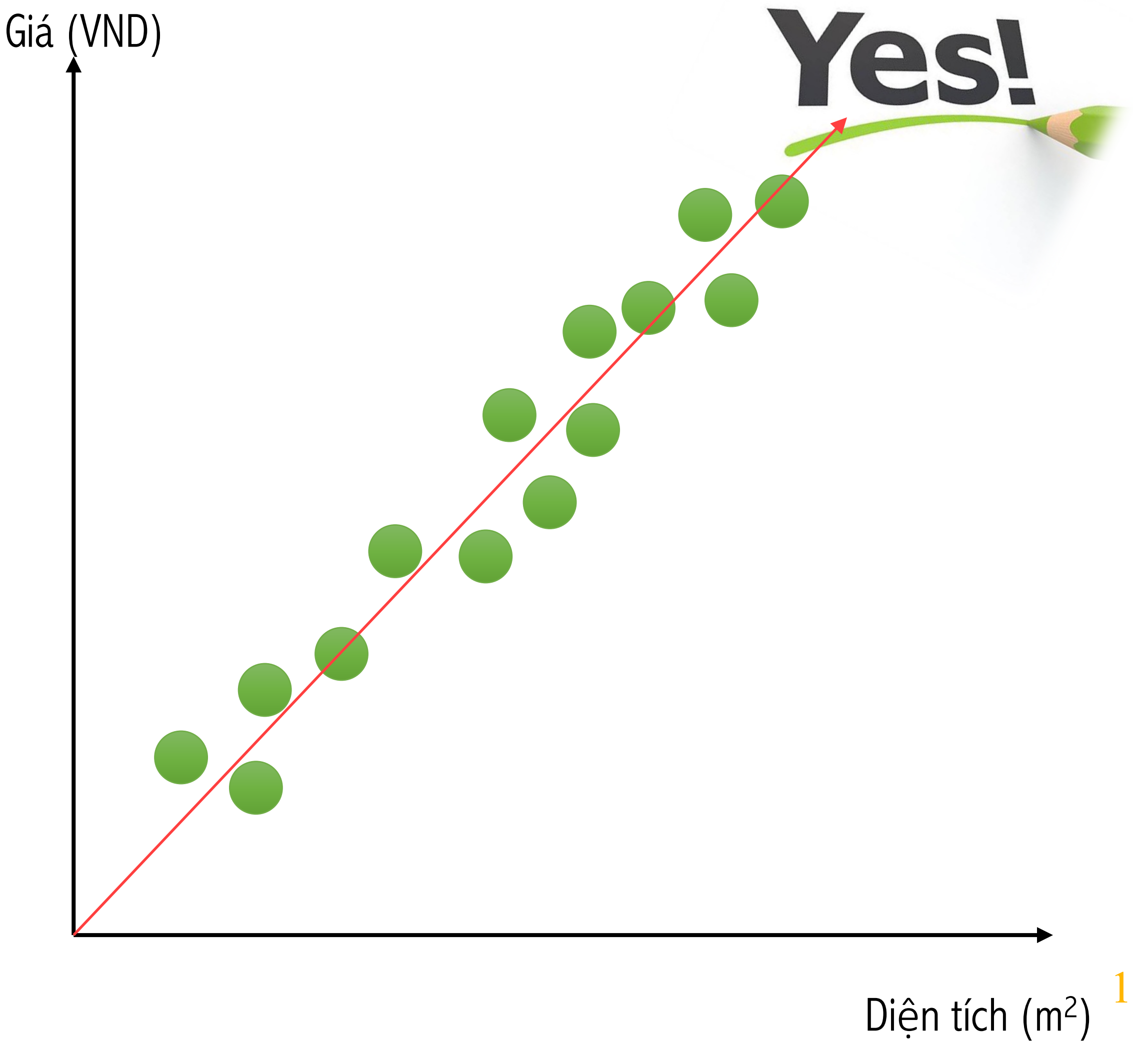
Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

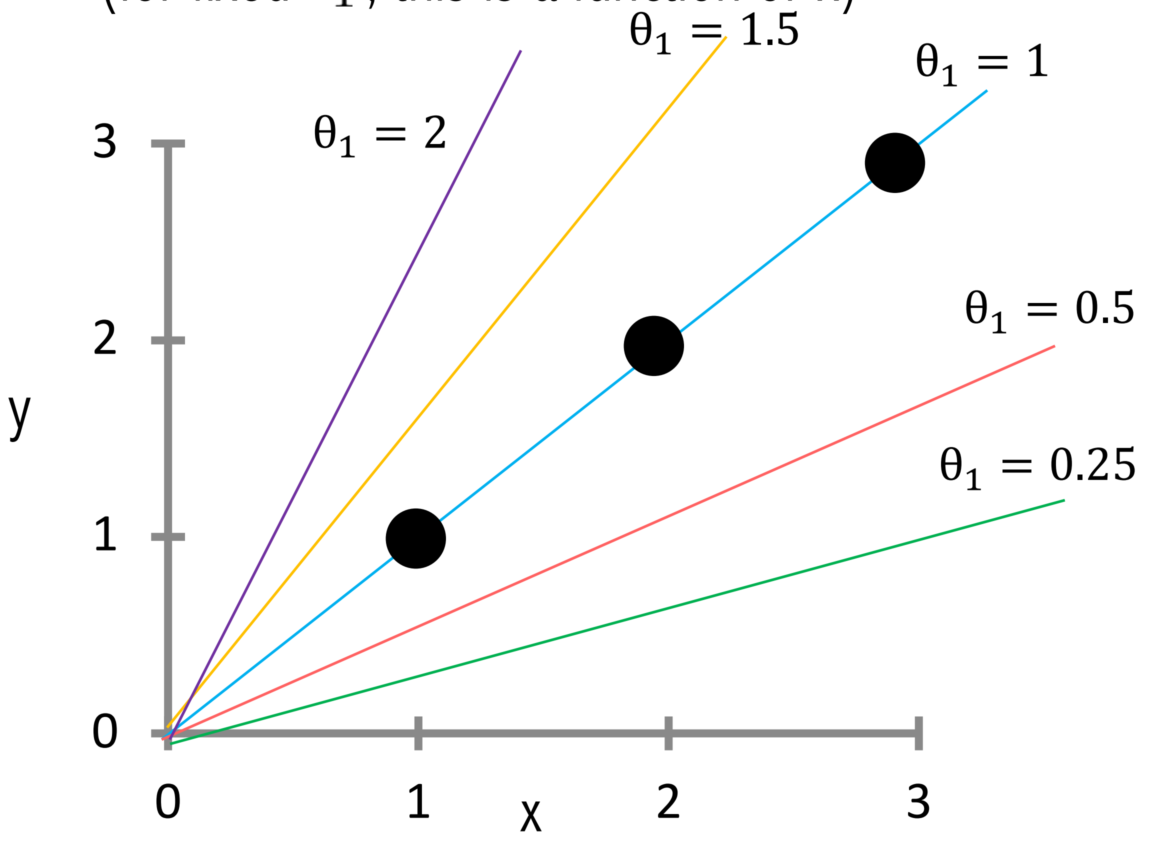
Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$



Linear Regression: Cost Function

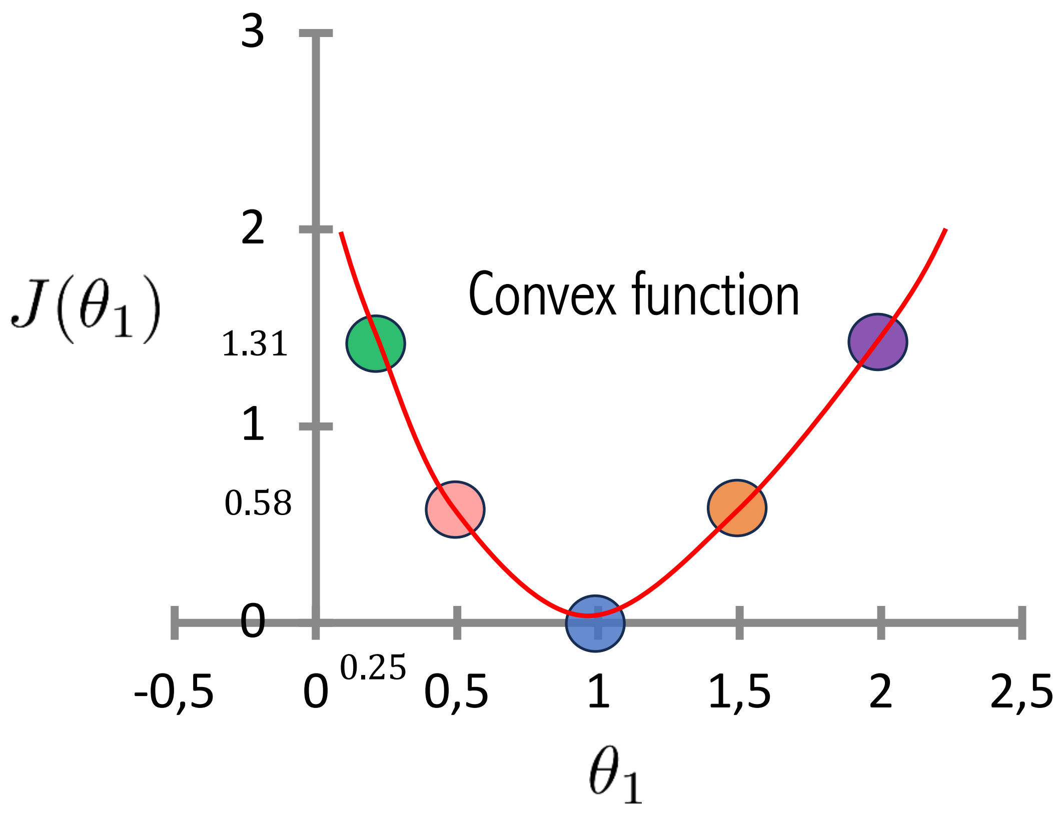
$h_\theta(x) = \theta_1 x$

(for fixed θ_1, this is a function of x)



$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

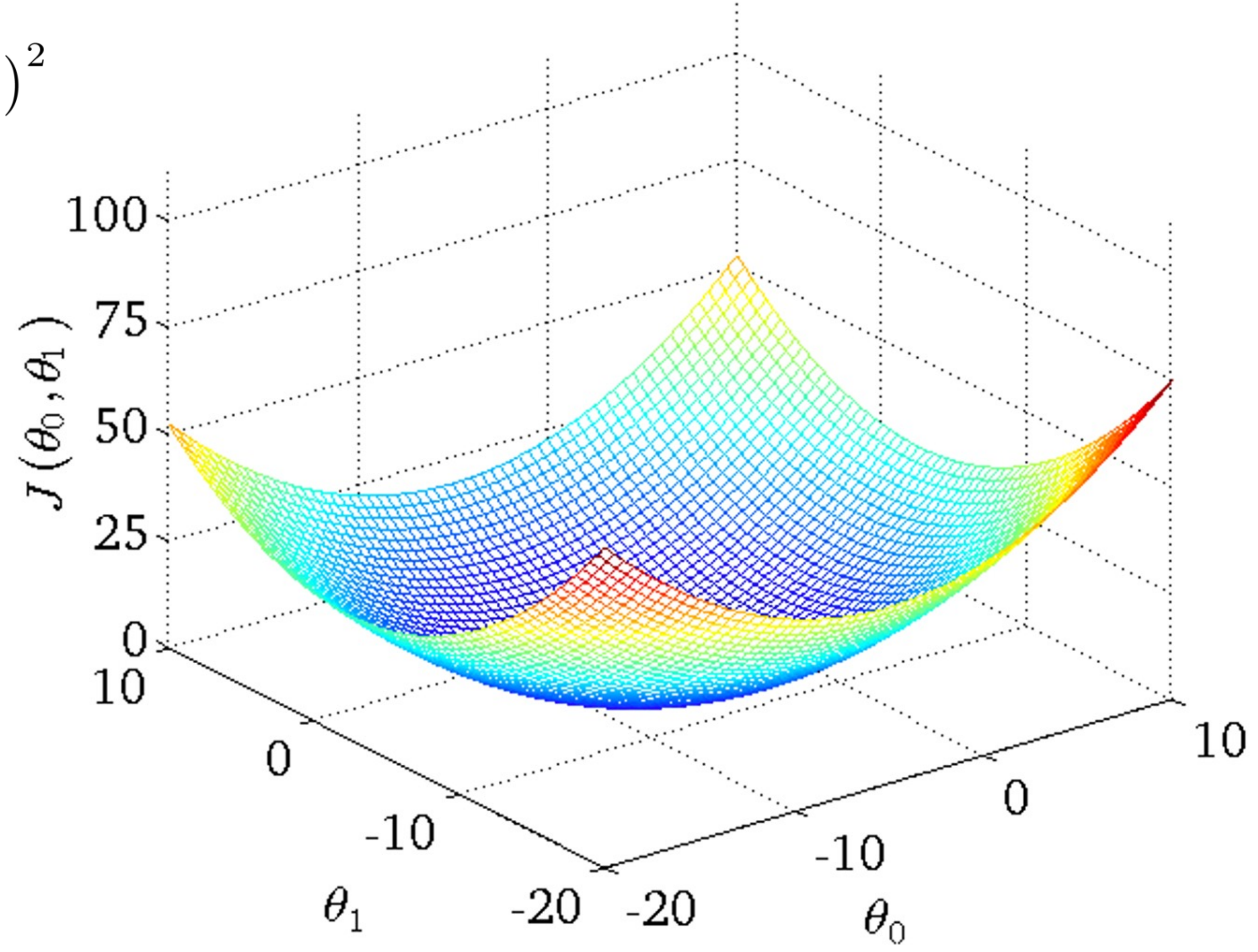
(function of the parameter θ_1)



Linear Regression: Cost Function

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$



Linear Regression: Cost Function

Hypothesis

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

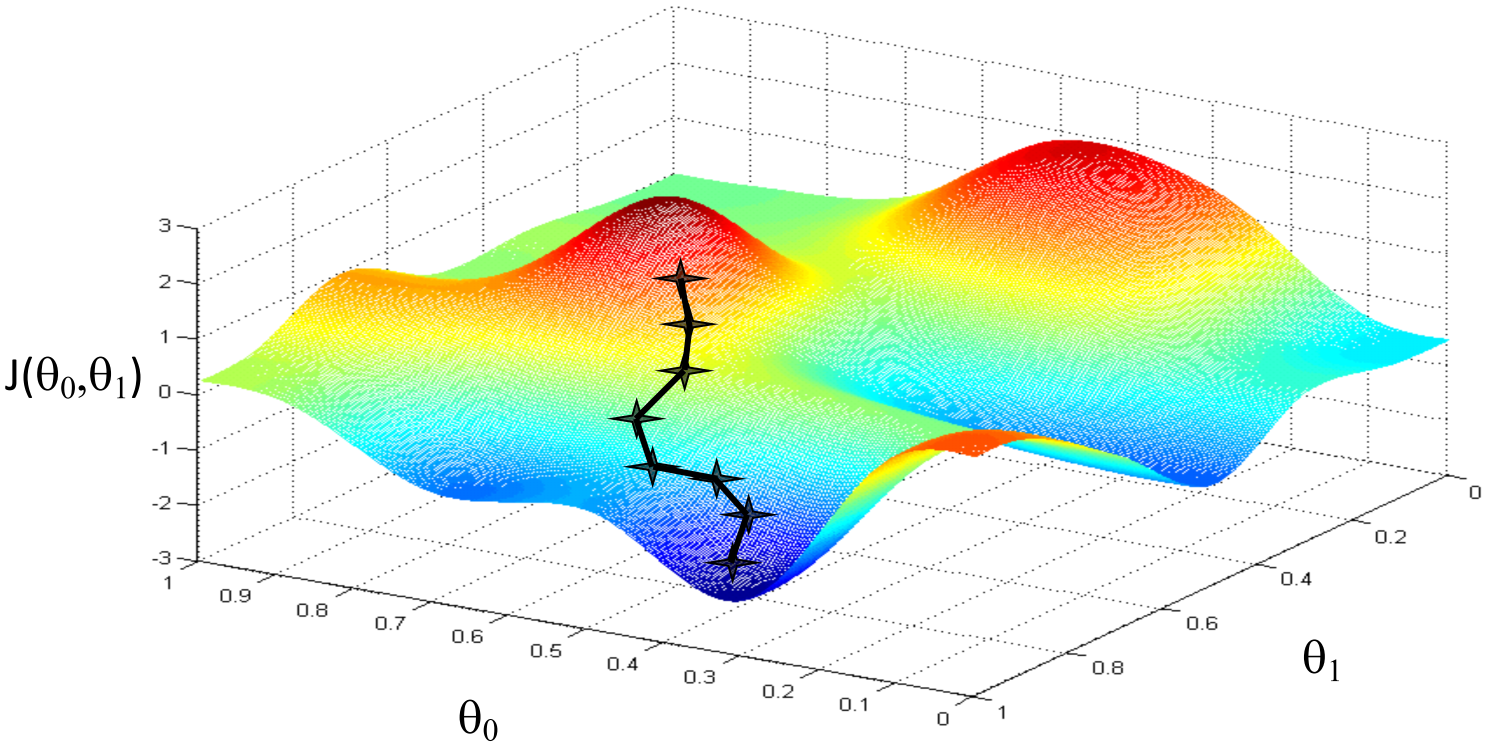
Gradient Descent

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$$

}



Linear Regression: Cost Function

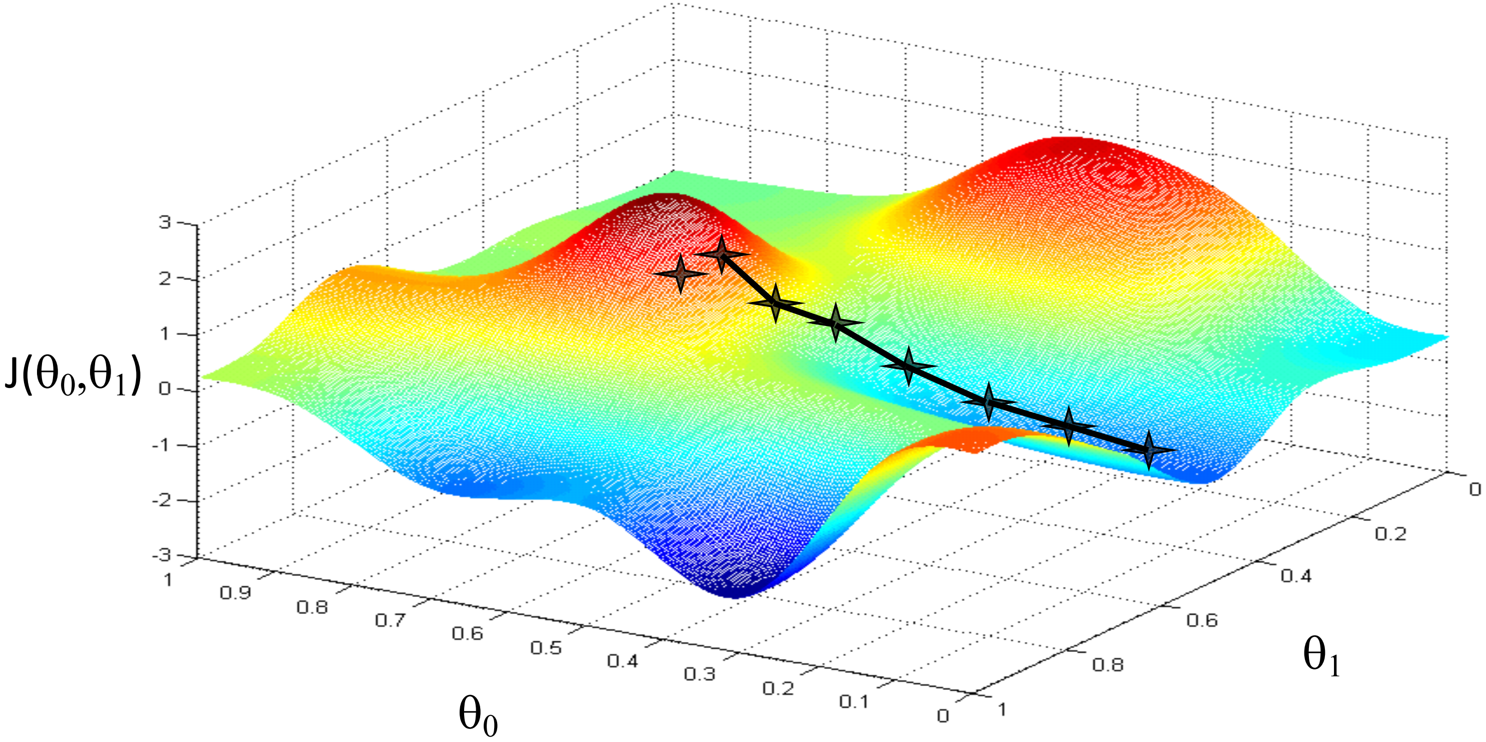
Hypothesis

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Gradient Descent

$$\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \\
&\quad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)} \\
&\}
\end{aligned}$$



Linear Regression for Classification

Develop a program to classify whether a tumor is malignant or not based on its size. Can we use Linear Regression to solve this problem?

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$h_\theta(x) = [-\infty, +\infty]$$

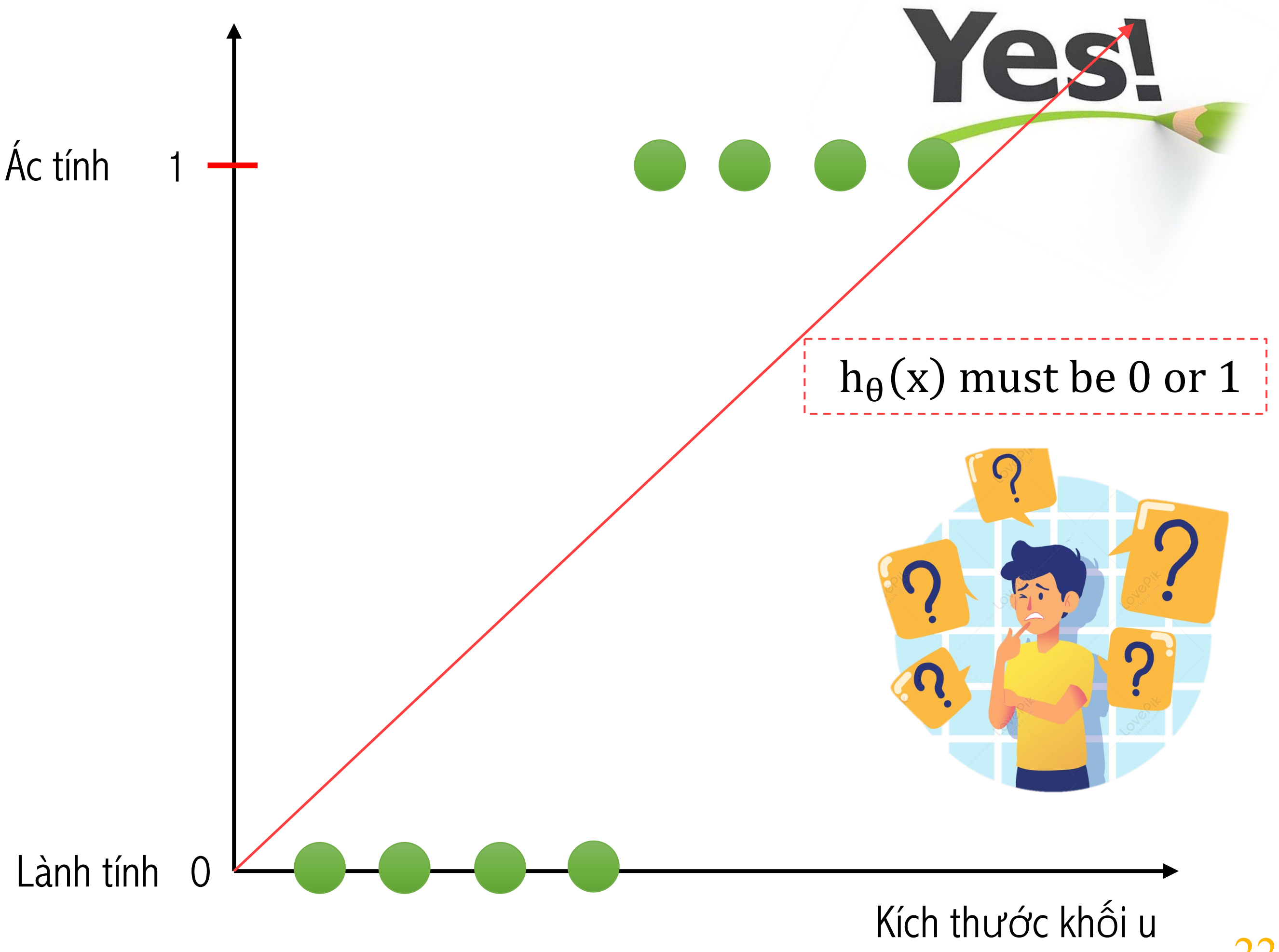
Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$



$h_\theta(x)$ must be 0 or 1

Linear Regression for Classification

Develop a program to classify whether a tumor is malignant or not based on its size. Can we use Linear Regression to solve this problem?

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$\boxed{h_\theta(x) = [-\infty, +\infty]}$$

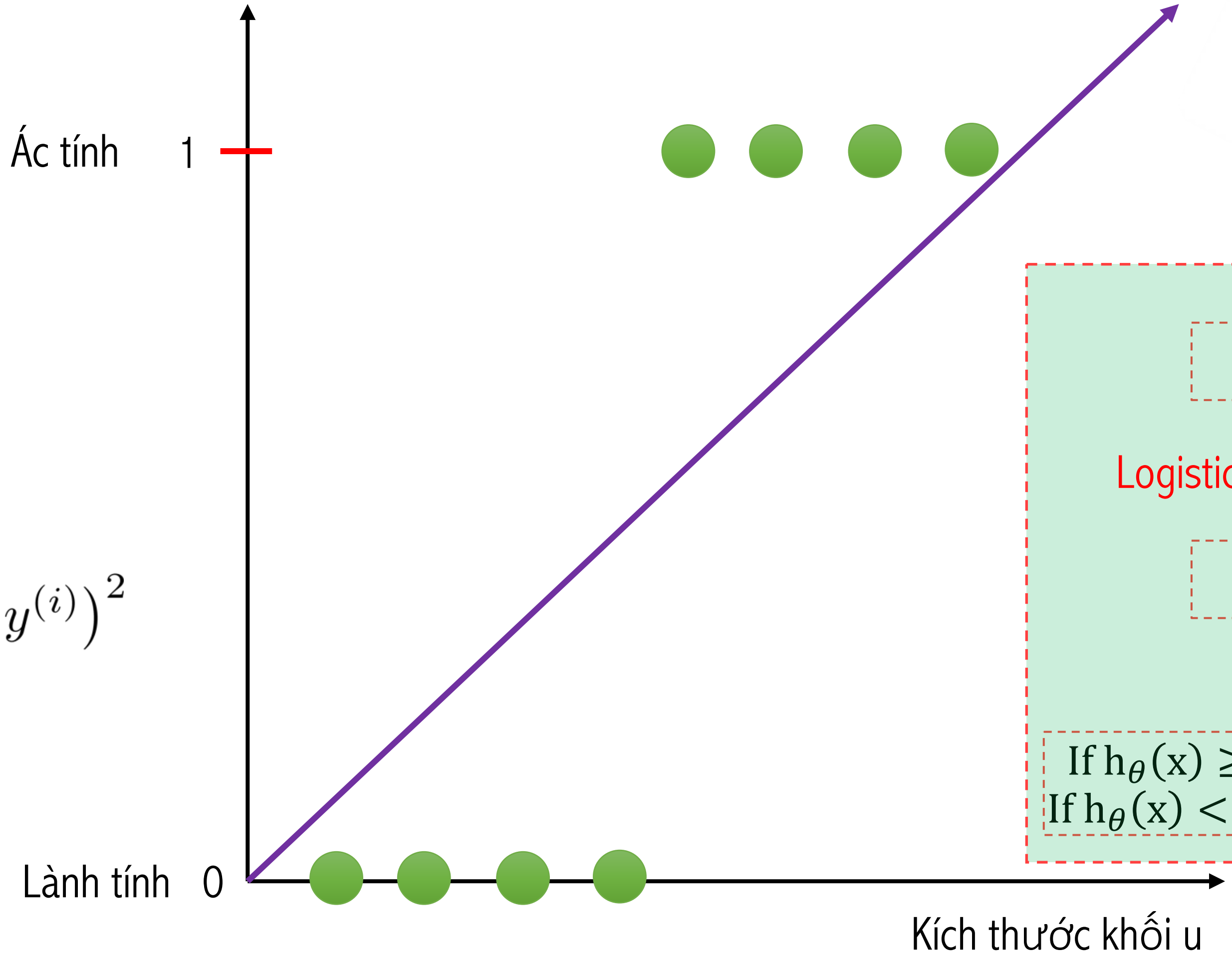
Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$



Logistic Regression

$$\boxed{h_\theta(x) = [-\infty, +\infty]}$$

Logistic function

$$\boxed{h_\theta(x) = [0,1]}$$

If $h_\theta(x) \geq 0.5$ predict 1 (ác tính)
If $h_\theta(x) < 0.5$ predict 0 (lành tính)

Logistic Regression for Classification

Hypothesis:

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$

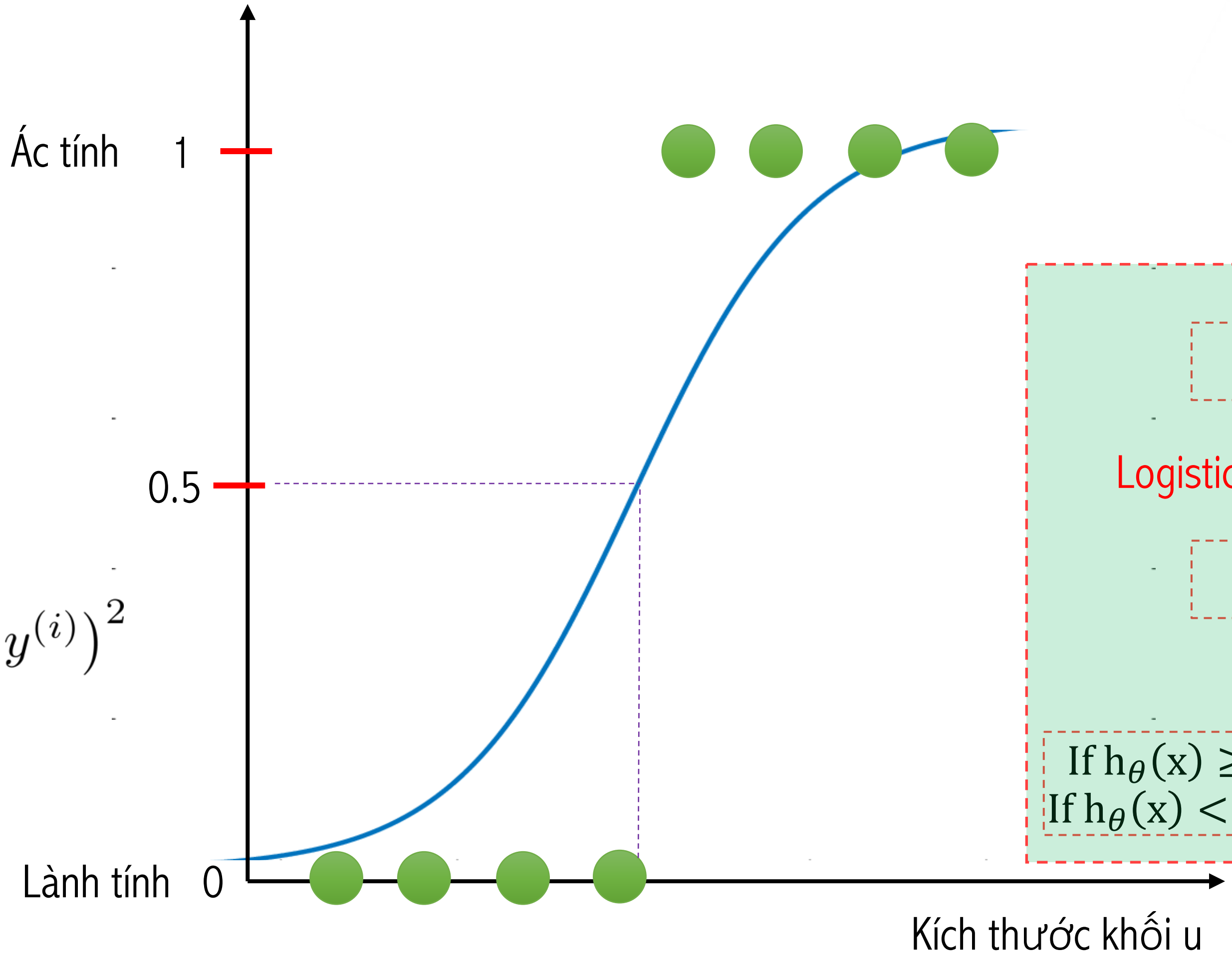
Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0,\theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$



Logistic Regression

$h_\theta(x) = [-\infty, +\infty]$

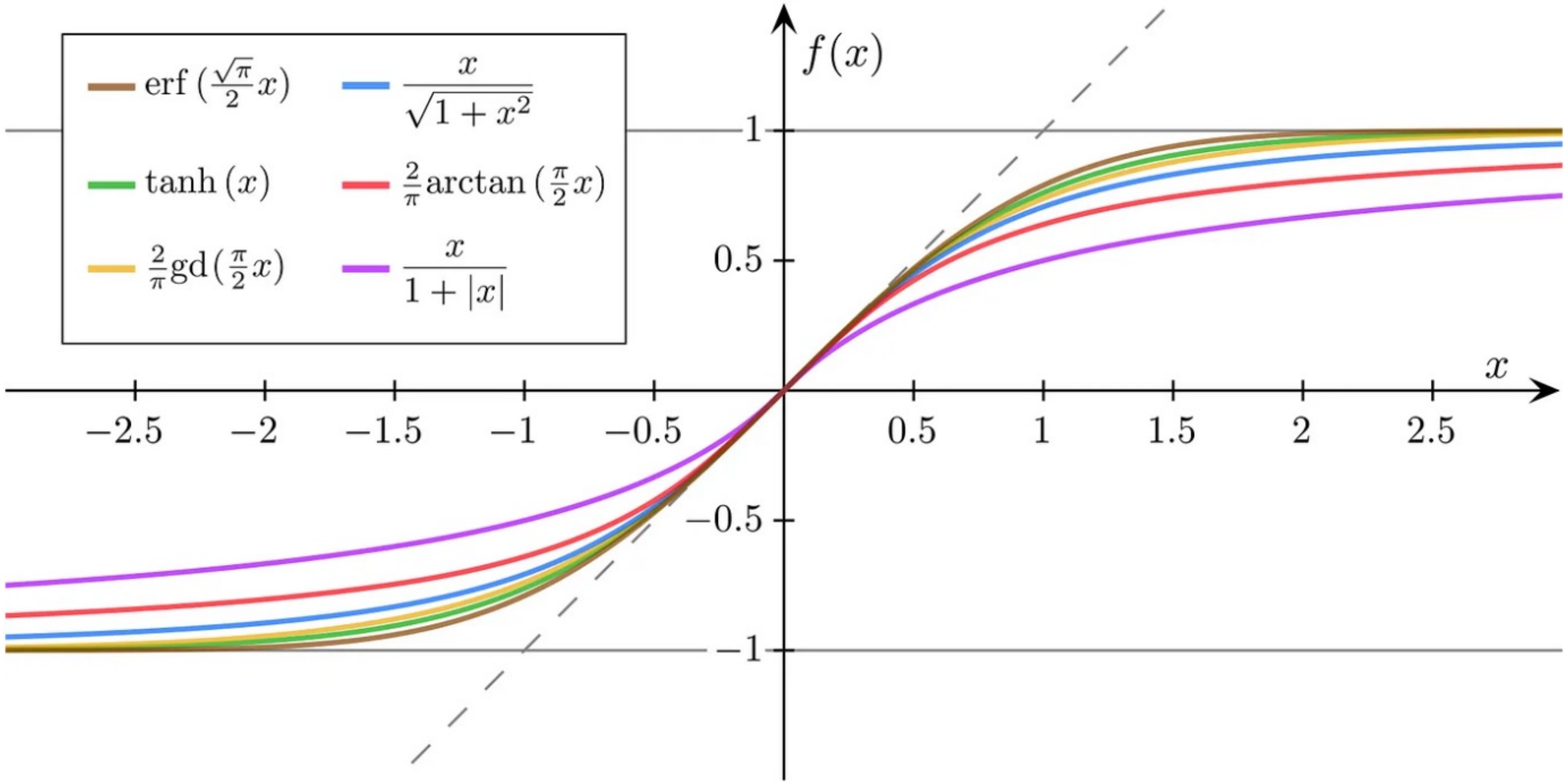
Logistic function

$h_\theta(x) = [0,1]$

If $h_\theta(x) \geq 0.5$ predict 1 (ác tính)
If $h_\theta(x) < 0.5$ predict 0 (lành tính)

Sigmoid Function

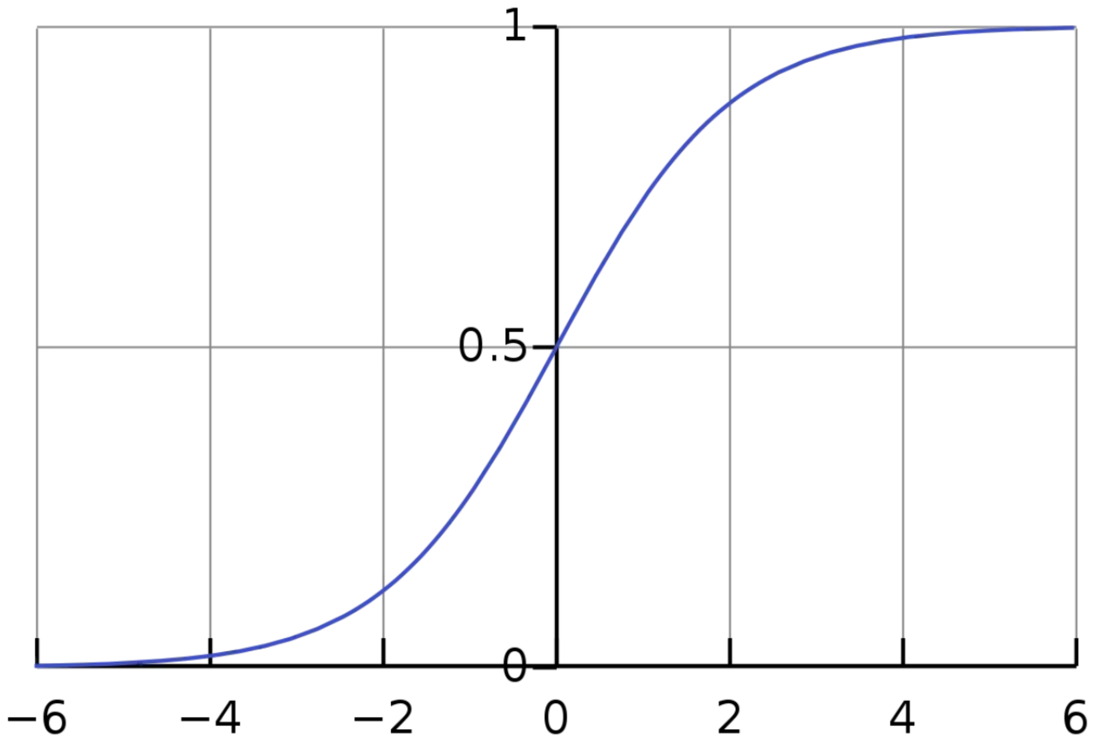
Sigmoid functions are general mathematical functions that share similar properties: have S-shaped curves, just as the figure below shows.



Members of Sigmoid Functions Family, from Wikipedia

Logistic Function

The most common sigmoid function used in machine learning is **Logistic Function**, as the formula below.



The Curve of a Logistic Function, from Wikipedia

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} = 1 - \sigma(-x).$$

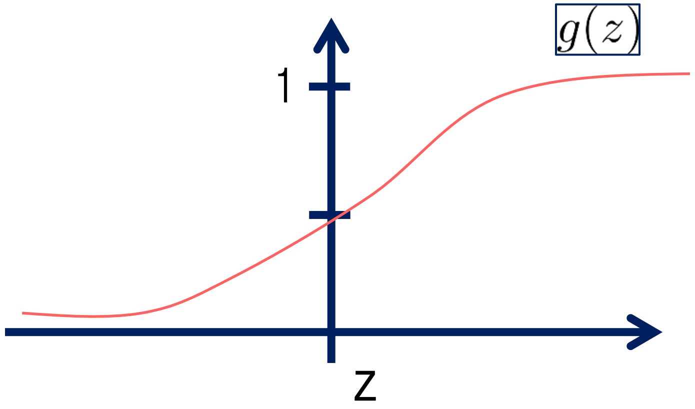
Logistic Regression

Hypothesis:

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$Z = \theta^T x \quad \theta = \begin{bmatrix}\theta_0 \\ \theta_1\end{bmatrix} \quad x = \begin{bmatrix}x_0 \\ x_1\end{bmatrix}$$



predict "$y = 1$" if $h_\theta(x) \geq 0.5$ ⇨ $g(z) \geq 0.5$ when $Z = \theta^T x \geq 0$

predict "$y = 0$" if $h_\theta(x) < 0.5$ ⇨ $g(z) < 0.5$ when $Z = \theta^T x < 0$

Cost Function

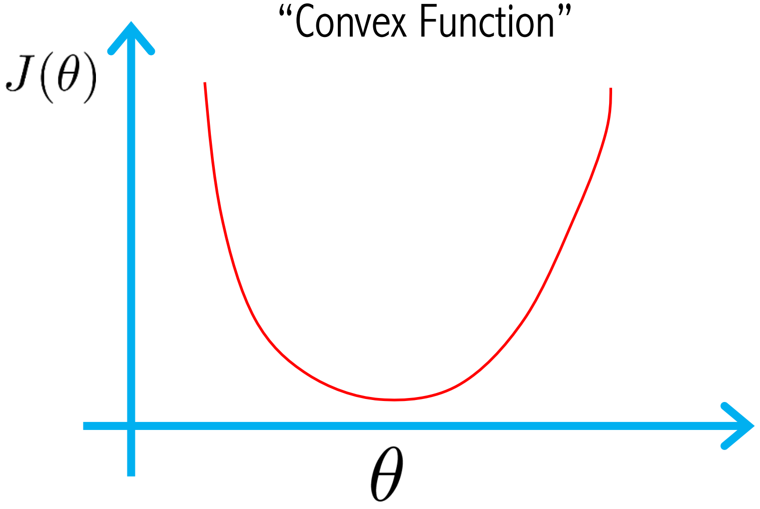
Linear Regression

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

"Convex Function"



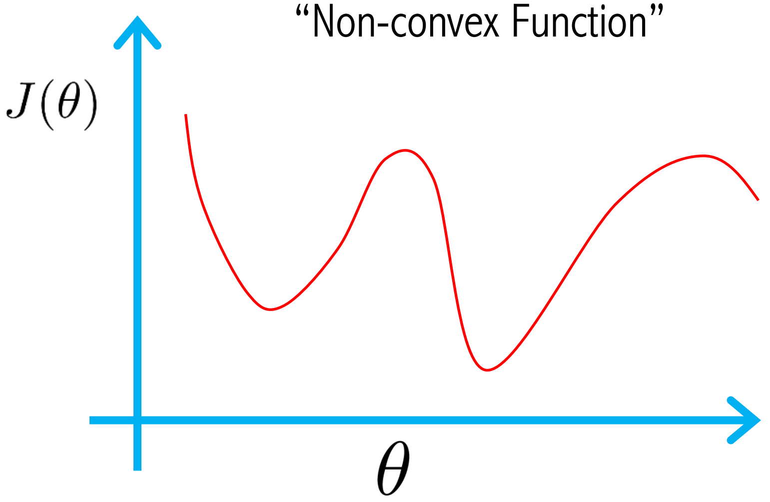
Logistic Regression

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

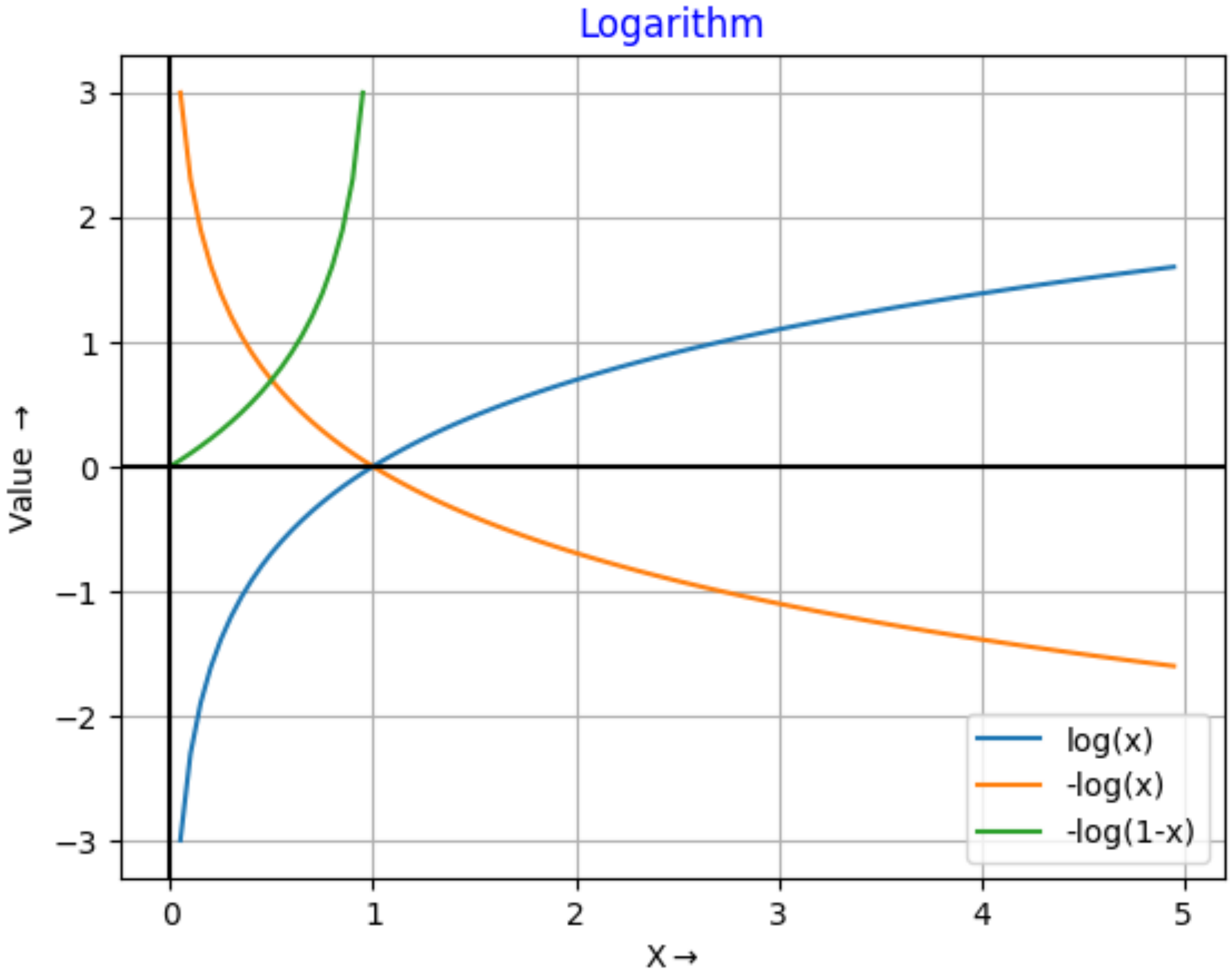
$$h_\theta(x) = g(\theta^T x) \quad g(z) = \frac{1}{1+e^{-z}}$$

$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

"Non-convex Function"



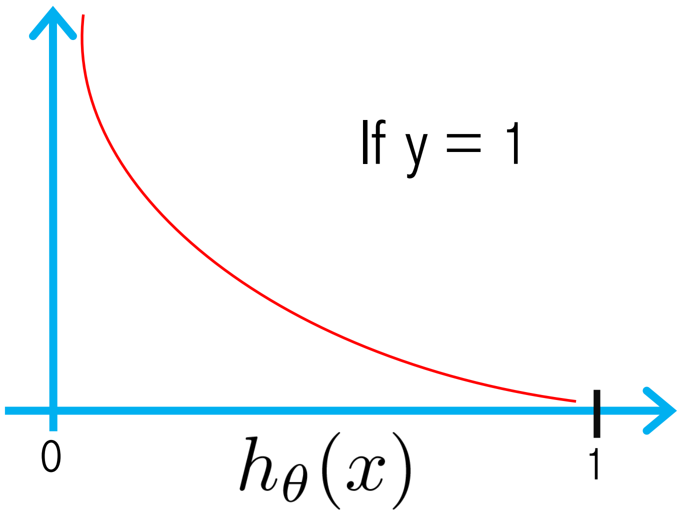
Logarithm Function



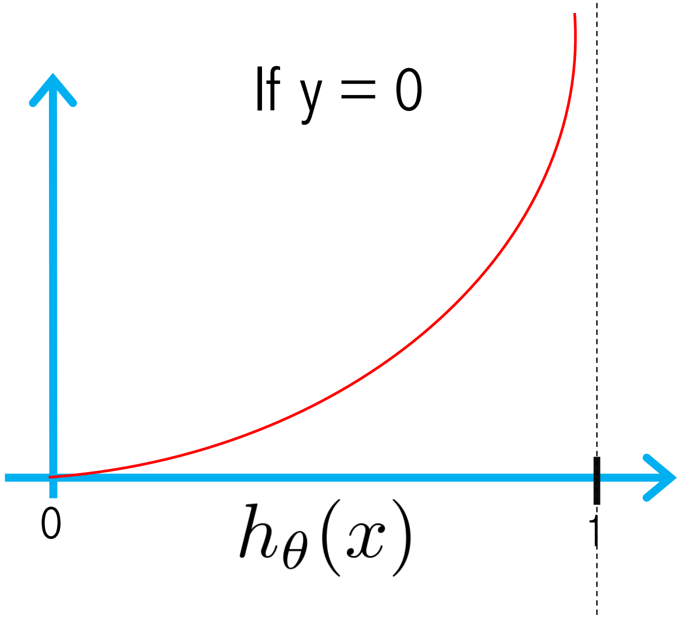
Logistic Regression: Cost Function

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$\boxed{\text{Cost} = 0 \text{ if } y = 1, h_\theta(x) = 1}$



$\boxed{\text{Cost} = 0 \text{ if } y = 0, h_\theta(x) = 0}$



Logistic Regression: Cost Function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$\mathrm{Cost}(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1 - h_\theta(x))$$

If $y = 1$: $\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x))$

If $y = 0$: $\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x))$

$$\begin{aligned} J(\theta) &= \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) \\ &= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))] \end{aligned}$$

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log h_\theta(x^{(i)})\right) + (1 - y^{(i)})\left((-\log(1 - h_\theta(x^{(i)}))\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Prevent Overfitting

Behind The Scene

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$\sigma = \frac{1}{(1+e^{-x})}$ $\sigma' = \sigma(x) \times (1 - \sigma(x))$ $z = \theta^T x$

For y = ln(u)

$\frac{dy}{dx} = \frac{1}{u}\frac{du}{dx}$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \times \frac{1}{h_\theta(x^{(i)})} \times \frac{\partial J\left(h_\theta(x^{(i)})\right)}{\partial \theta_j}\right] + \frac{1}{m}\sum_{i=1}^{m}\left[(1 - y^{(i)}) \times \frac{1}{\left(1 - h_\theta(x^{(i)})\right)} \times \frac{\partial J\left(1 - h_\theta(x^{(i)})\right)}{\partial \theta_j}\right]$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \times \frac{1}{h_\theta(x^{(i)})} \times (\sigma(z)(1 - \sigma(z)) \times \frac{\partial J(z)}{\partial \theta_j}\right] + \frac{1}{m}\sum_{i=1}^{m}\left[(1 - y^{(i)}) \times \frac{1}{\left(1 - h_\theta(x^{(i)})\right)} \times (-\sigma(z)(1 - \sigma(z)) \times \frac{\partial J(z)}{\partial \theta_j}\right]$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \times \frac{1}{h_\theta(x^{(i)})} \times (h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})) \times x_j^i)\right] + \frac{1}{m}\sum_{i=1}^{m}\left[(1 - y^{(i)}) \times \frac{1}{\left(1 - h_\theta(x^{(i)})\right)} \times (-h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})) \times x_j^i\right]$$

Behind The Scene

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$\sigma = \frac{1}{(1+e^{-x})}$ $\sigma' = \sigma(x) \times (1 - \sigma(x))$ $z = \theta^T x$

For y = ln(u)

$\frac{dy}{dx} = \frac{1}{u}\frac{du}{dx}$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \times (1 - h_\theta\left(x^{(i)}\right) \times x_j^i - \left(1 - y^{(i)}\right) \times h_\theta\left(x^{(i)}\right) \times x_j^i\right]$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} - y^{(i)} \times h_\theta\left(x^{(i)}\right) - h_\theta\left(x^{(i)}\right) + y^{(i)} \times h_\theta\left(x^{(i)}\right)\right] \times x_j^i$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} - h_\theta\left(x^{(i)}\right)\right] \times x_j^i$$

The gradient descent is the same as Linear Regression

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\sigma = \frac{1}{(1+e^{-x})}$$

Outline

- Linear Regression to Logistic Regression
- Logistic Regression to Support Vector Machine
- Example

Another View of Logistic Regression

Hypothesis:

$$h_\theta(x) = g(\theta^T x)$$

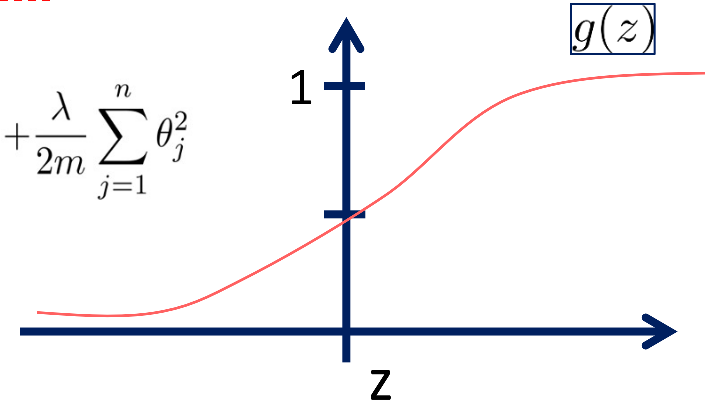
$$g(z) = \frac{1}{1+e^{-z}}$$

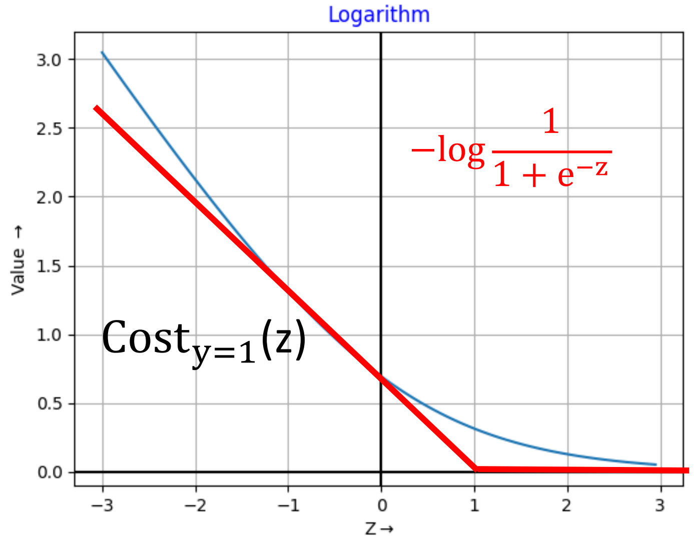
$$Z = \theta^T x$$

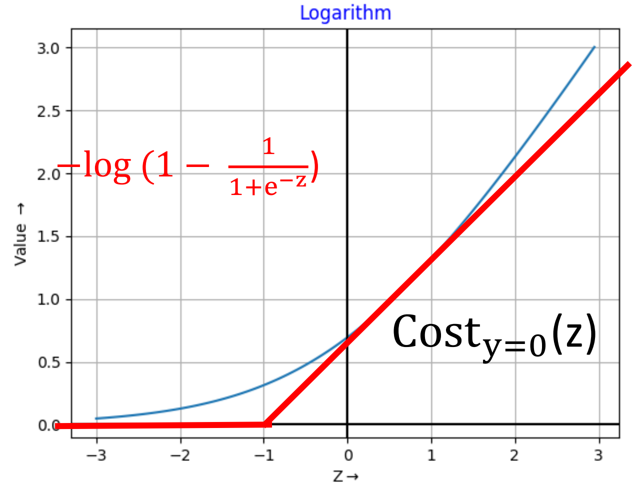
$\mathrm{Cost}_{y=1}(\theta^T x^i)$ $\quad$ $\mathrm{Cost}_{y=0}(\theta^T x^i)$

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \boxed{\left(-\log h_\theta(x^{(i)})\right)} + (1-y^{(i)})\left(\boxed{\left(-\log(1-h_\theta(x^{(i)}))\right)}\right.\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$\boxed{\mathrm{Cost}(h_\theta(x), y) = -y\log\big(h_\theta(x)\big) - (1-y)\log\big(1 - h_\theta(x)\big)}$$







If $y = 1$, we want $h_\theta(x) \approx 1, Z = \theta^T x \gg 0$

If $y = 0$, we want $h_\theta(x) \approx 0, Z = \theta^T x \ll 0$

Logistic Regression to Support Vector Machine

Logistic Regression

$\text{Cost}_{y=1}(\theta^T x^i)$ $\text{Cost}_{y=0}(\theta^T x^i)$

Sắp xỉ Sắp xỉ

$$\min_\theta \underbrace{\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(\boxed{-\log h_\theta(x^{(i)})}\right) + (1-y^{(i)})\boxed{\left((-\log(1-h_\theta(x^{(i)}))\right)}\right]}_{A} + \frac{\lambda}{2m}\underbrace{\sum_{j=1}^{n}\theta_j^2}_{B}$$

A B

$\min (A + \lambda B)$

Support Vector Machine

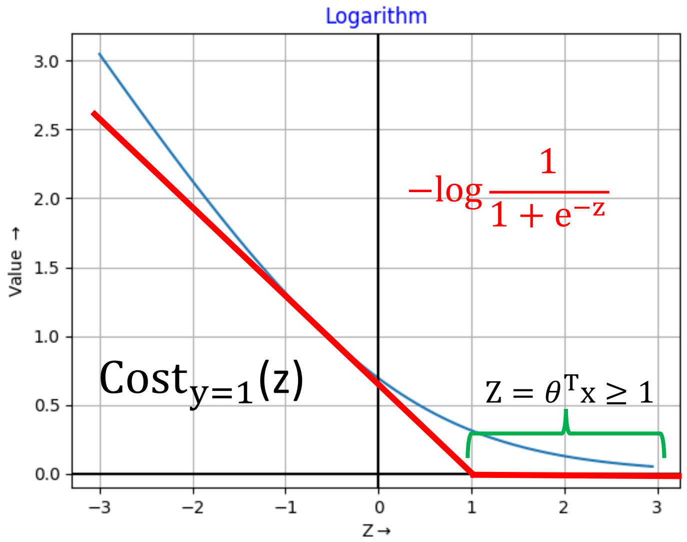
$\min (CA + B)$ $C = 1 / \lambda$

A B

$$\min_\theta C \underbrace{\sum_{i=1}^{m} -y_i\text{Cost}_{y=1}(\theta^T x^i) + (1-y)\text{Cost}_{y=0}(\theta^T x^i)}_{A} + \underbrace{\frac{1}{2}\sum_{j=1}^{m}\theta_j^2}_{B}$$

Support Vector Machine

$$\min_{\theta} C \sum_{i=1}^{m} -yiCost_{y=1}(\theta^T x^i) + (1-y)Cost_{y=0}(\theta^T x^i) + \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$



Logistic Regression

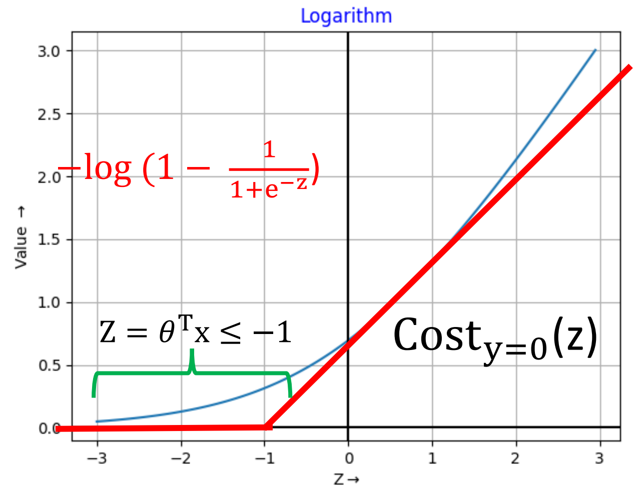
If $y = 1$, we want $h_\theta(x) \approx 1, Z = \theta^T x > 0$

Support Vector Machine

If $y = 1$, we want $h_\theta(x) \approx 1, Z = \theta^T x \geq 1$

Supposing that C is very very large (100,000)

Does it affect to the optimization?



Logistic Regression

If $y = 1$, we want $h_\theta(x) \approx 0, Z = \theta^T x < 0$

Support Vector Machine

If $y = 1$, we want $h_\theta(x) \approx 1, Z = \theta^T x \leq -1$

SVM Decision Boundary

Supposing that C is very very large (100,000)

1

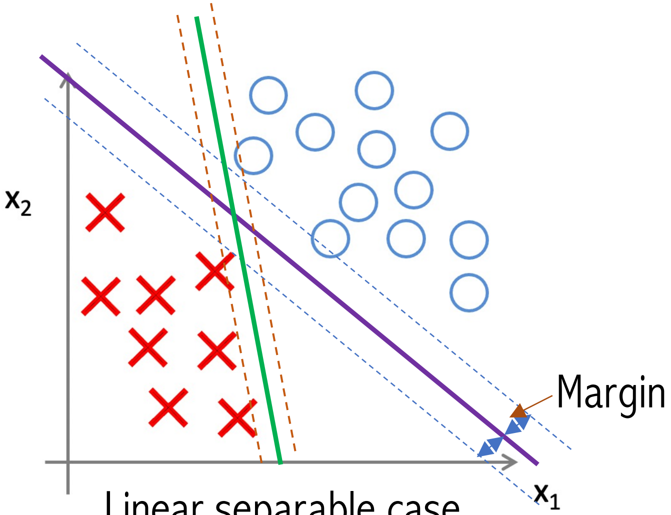
This term should be ….

2

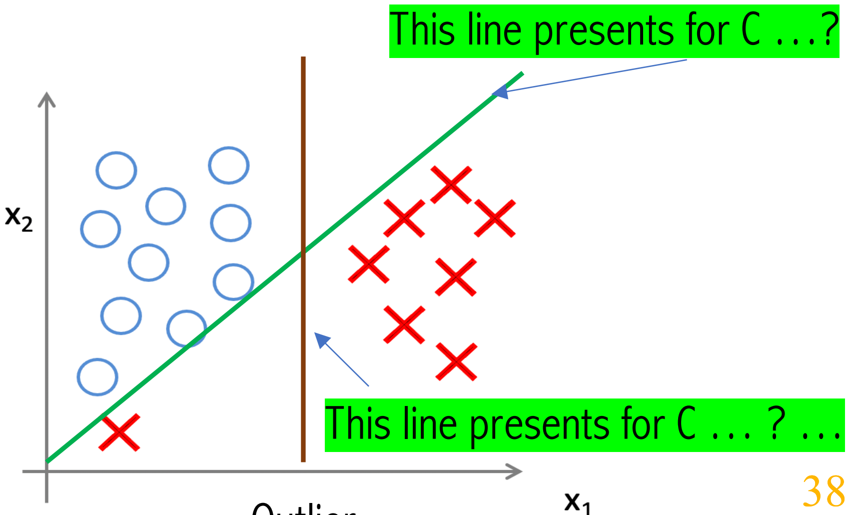
$$\min \boxed{C} \sum_{j=1}^{m} -yiCost_{y=1}(\theta^T x^i) + (1-y)Cost_{y=0}(\theta^T x^i) + \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$

3

$$\min C \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$

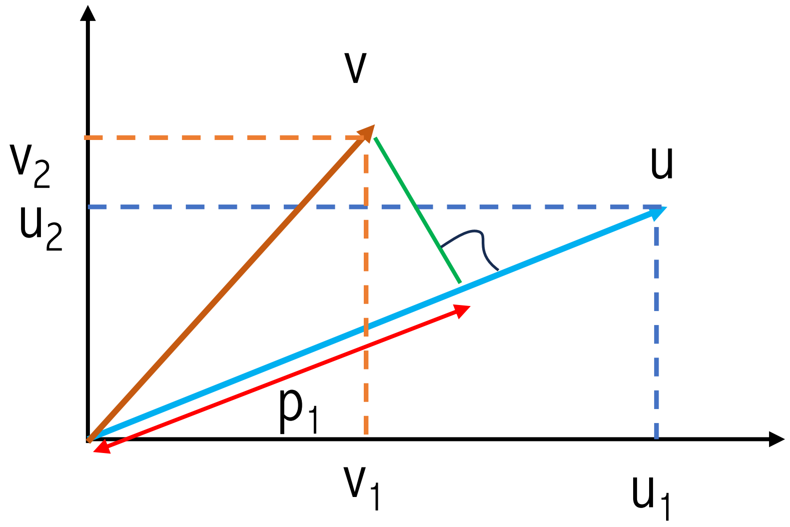


Linear separable case



Outlier

Vector Innder Product

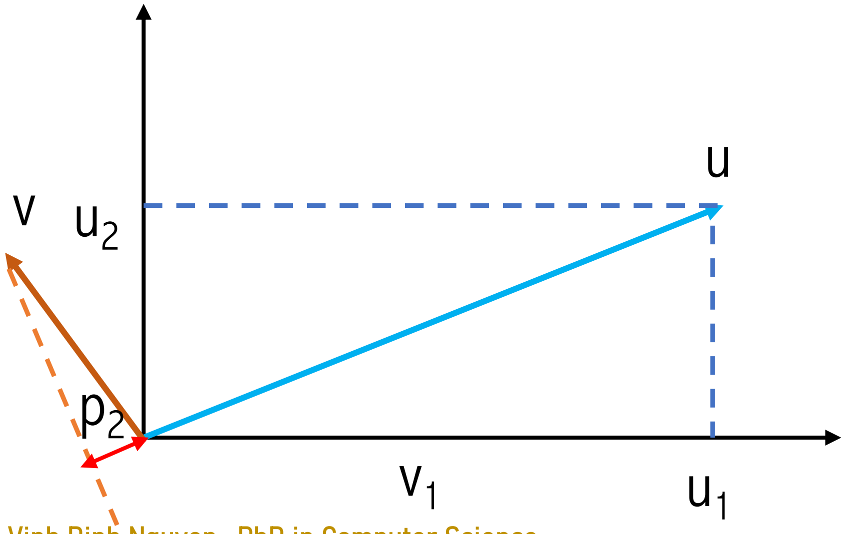


$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ $\quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$

Length of vector u $\|u\| = \sqrt{u_1^2 + u_2^2}$

$u^T v = p_1 . \|u\| = u_1 v_1 + u_2 v_2$

p_1: Length of projection of vector v on vector u



$u^T v = p_2 . \|u\| = u_1 v_1 + u_2 v_2$

What are the value of p in two cases?

a. $p_1 > 0$ & $p_2 < 0$
b. $p_1 < 0$ & $p_2 > 0$
c. $p_1 > 0$ & $p_2 > 0$
d. $p_1 < 0$ & $p_2 < 0$

SVM Decision Boundary

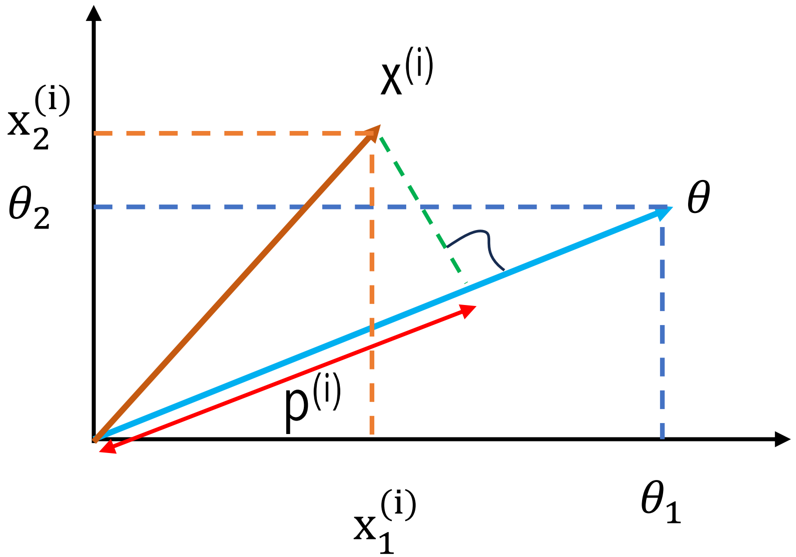
$$\min C\frac{1}{2}\sum_{j=1}^{m}\theta_j^2 = \frac{1}{2}(\theta_0^2+\theta_1^2+\theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_0^2+\theta_1^2+\theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2$$

Subjective to

$$\theta^T x^{(i)} \geq 1 \qquad \text{If } y^{(i)} = 1$$
$$\theta^T x^{(i)} \leq -1 \qquad \text{If } y^{(i)} = 0$$

Simplication:

$$\theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix} \quad \theta_0 = 0 \quad n = 2 \quad x^{(i)} = \begin{bmatrix}x_0^{(i)}\\ x_1^{(i)}\\ x_2^{(i)}\end{bmatrix} \quad x_0^{(i)} = 1$$



$$\theta^T x^{(i)} = p^{(i)}\|\theta\| = \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$$

SVM Decision Boundary

$$\min C\frac{1}{2}\sum_{j=1}^{m}\theta_j^2 = \frac{1}{2}(\theta_0^2+\theta_1^2+\theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_0^2+\theta_1^2+\theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2 \quad (1)$$

Subjective to

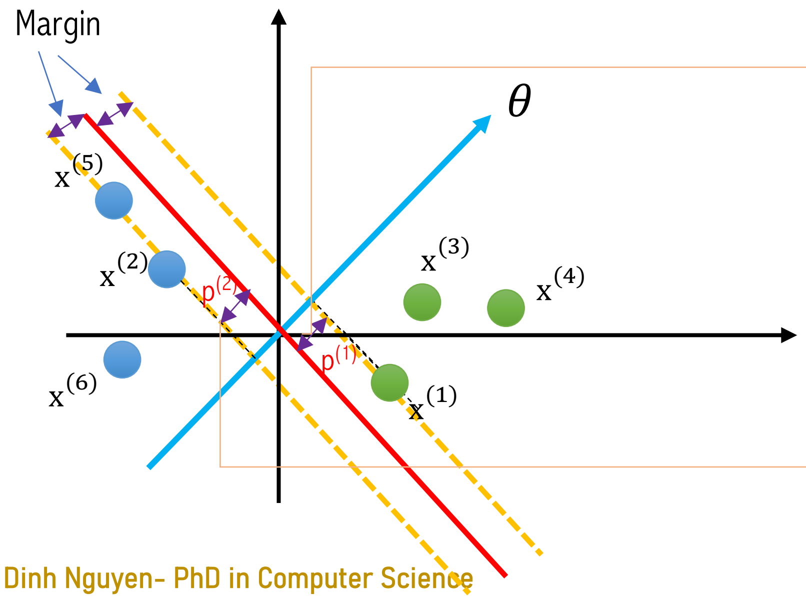
$p^{(i)}\|\theta\| \geq 1$ If $y^{(i)} = 1$

$p^{(i)}\|\theta\| \leq -1$ If $y^{(i)} = 0$

$p^{(i)}$ is the projection of $\mathbf{x}^{(i)}$ of onto the vector θ

Simplication:

$$\theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix} \quad \theta_0 = 0 \quad n = 2 \quad \mathbf{x}^{(i)} = \begin{bmatrix}x_0^{(i)}\\ x_1^{(i)}\\ x_2^{(i)}\end{bmatrix} \quad x_0^{(i)} = 1$$



$p^{(i)}\|\theta\| \geq 1$

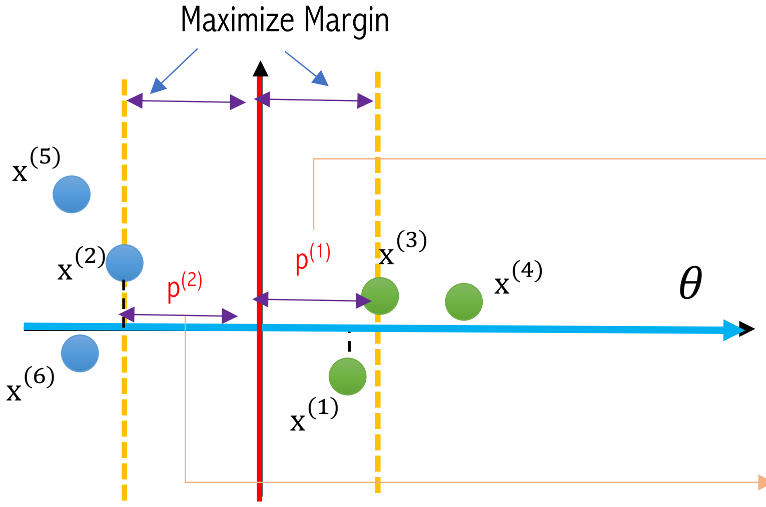
$\|\theta\|$ is large

Violate (1)

$p^{(i)}\|\theta\| \leq -1$

$\|\theta\|$ is large

Violate (1)



$p^{(i)}\|\theta\| \geq 1$

$\|\theta\|$ is small

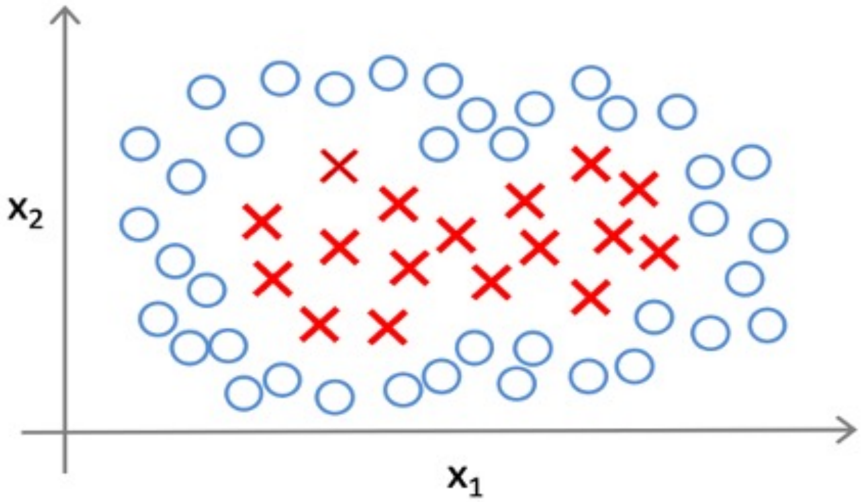
Satisfy (1)

$p^{(i)}\|\theta\| \leq -1$

$\|\theta\|$ is small

Satisfy (1)

Non-Linear Decision Boundary



Có phương pháp nào để xây dựng features $f_1, f_2, f_3,$
tố hơn không không ?

Linear case

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_3 x_4 + \theta_5 x_5 + \cdots$$

$$h_\theta(x) = \begin{cases} 1 & z \geq 1 \\ 0 & z \leq -1 \end{cases}$$

Polynomial case

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots$$

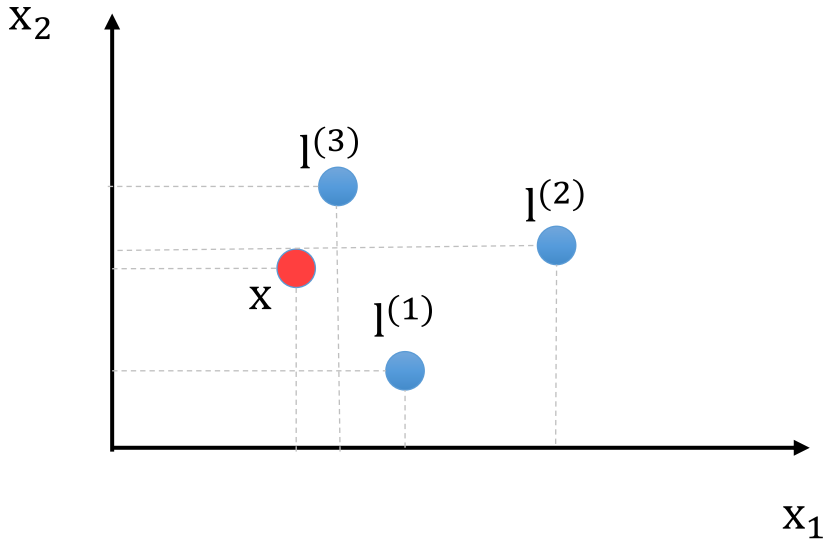
$$h_\theta(x) = \begin{cases} 1 & z \geq 1 \\ 0 & z \leq -1 \end{cases}$$

Polynomial case

$$z = \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_2 f_3 + \theta_4 f_4 \ldots$$

$$f_1 = x_1,\ f_2 = x_2,\ f_3 = x_1 x_2,\ f_4 = x_1^2,\ f_5 = x_2^2,\ \ldots$$

Kernel and Similarity



Given sample x $[x_1, x_2]$, compute new feature depending on proximity to landmarks $l^{(1)}, l^{(2)}, l^{(3)}$

Generate new feature

$$f_1 = \text{similarity}\left(x, l^{(1)}\right) = e^{\left(-\frac{\left\|x - l^{(1)}\right\|^2}{2\sigma^2}\right)}$$

$$f_2 = \text{similarity}\left(x, l^{(2)}\right) = e^{\left(-\frac{\left\|x - l^{(2)}\right\|^2}{2\sigma^2}\right)}$$

$$f_3 = \text{similarity}\left(x, l^{(3)}\right) = e^{\left(-\frac{\left\|x - l^{(3)}\right\|^2}{2\sigma^2}\right)}$$

Kernel

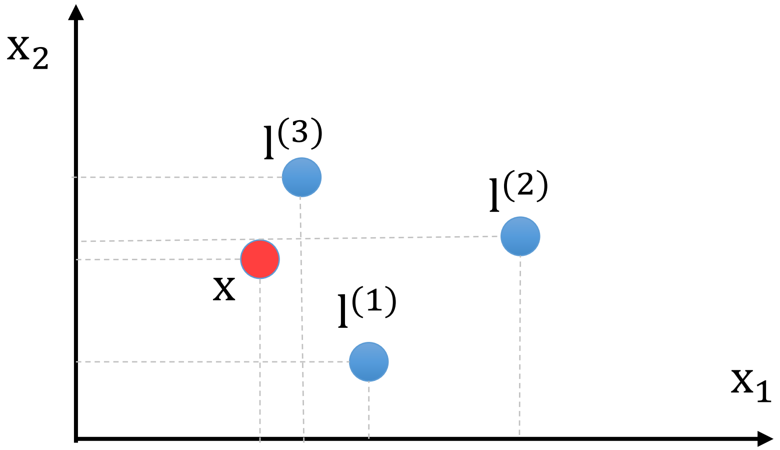
Gausian kernel

X and $l^{(1)}$ Relationship

If x is near to $\approx l^{(1)}$: $f_1 = e^{\left(-\frac{\left\|x - l^{(1)}\right\|^2}{2\sigma^2}\right)} \approx 1$

If is far from $l^{(1)}$: $f_1 = e^{\left(-\frac{\left\|x - l^{(1)}\right\|^2}{2\sigma^2}\right)} \approx 0$

Landmark Selection



Given samples: $\left((x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\right)$

Choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \dots, l^{(m)} = x^{(m)}$

Old feature

$$x_{old}^{(i)} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Generate new features

Given sample x:

$$f_1^{(i)} = \text{similarity}\left(x^{(i)}, l^{(1)}\right) = e^{\left(-\frac{\left\|x^{(i)} - l^{(1)}\right\|^2}{2\sigma^2}\right)}$$

$$f_2^{(i)} = \text{similarity}\left(x^{(i)}, l^{(2)}\right) = e^{\left(-\frac{\left\|x^{(i)} - l^{(2)}\right\|^2}{2\sigma^2}\right)}$$

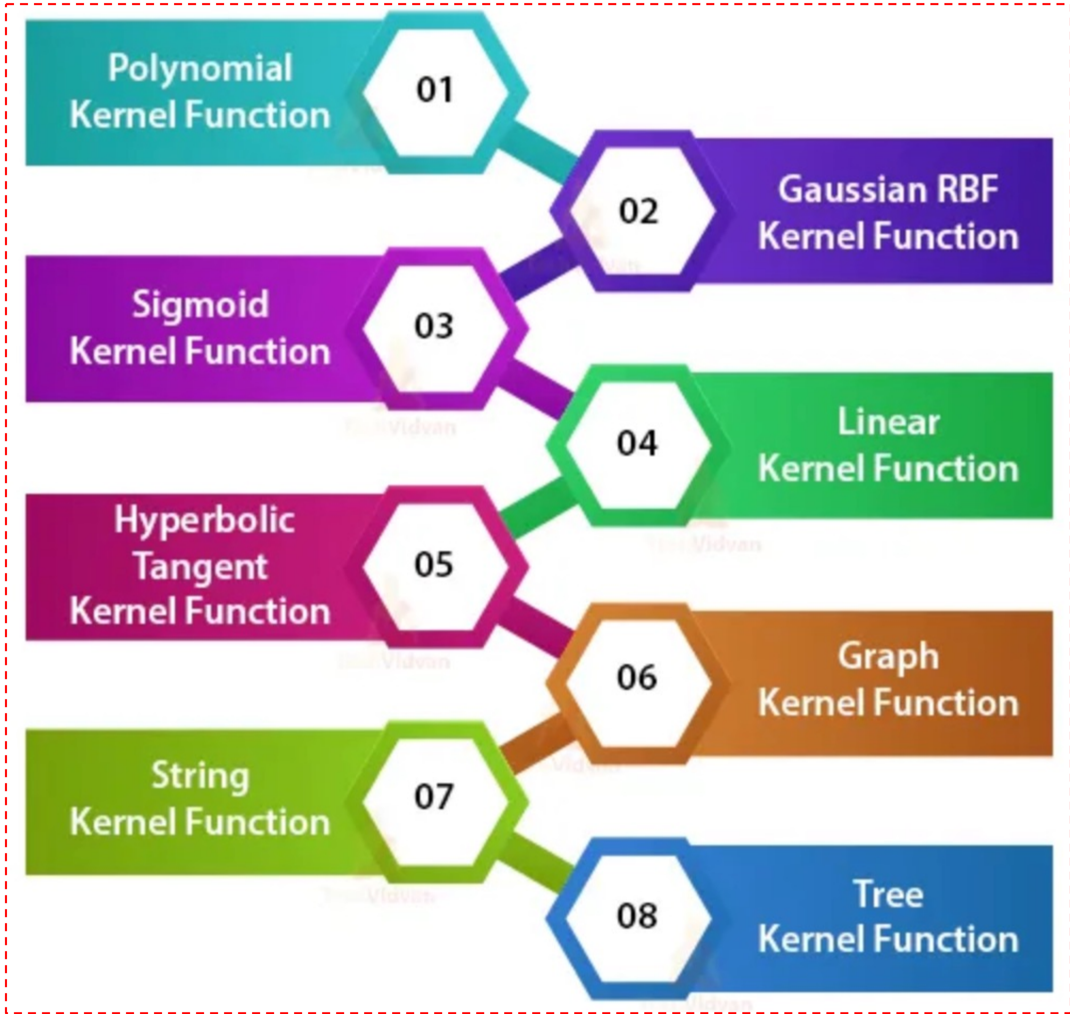
...

$$f_m^{(i)} = \text{similarity}\left(x^{(i)}, l^{(m)}\right) = e^{\left(-\frac{\left\|x^{(i)} - l^{(m)}\right\|^2}{2\sigma^2}\right)}$$

New feature

$$x_{new}^{(i)} = f = \begin{bmatrix} f_1^{(i)} \\ f_2^{(i)} \\ f_3^{(i)} \\ \dots \\ f_m^{(i)} \end{bmatrix}$$

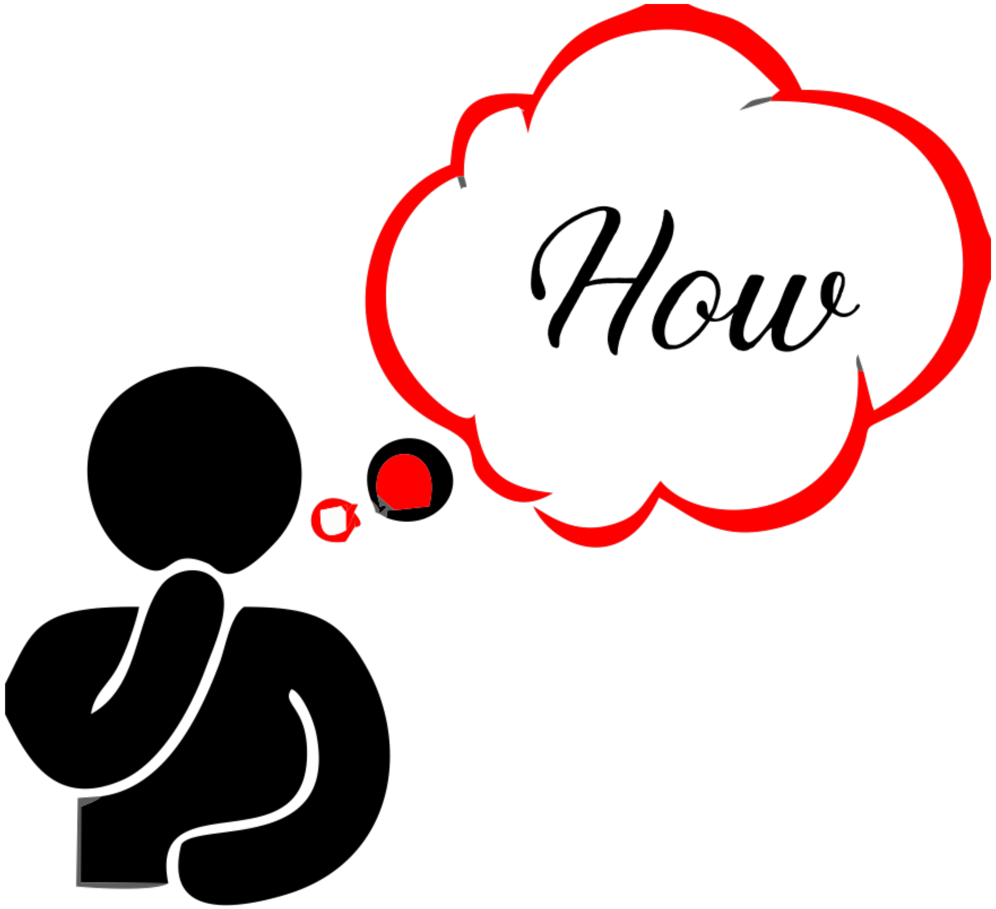
Other Kernel

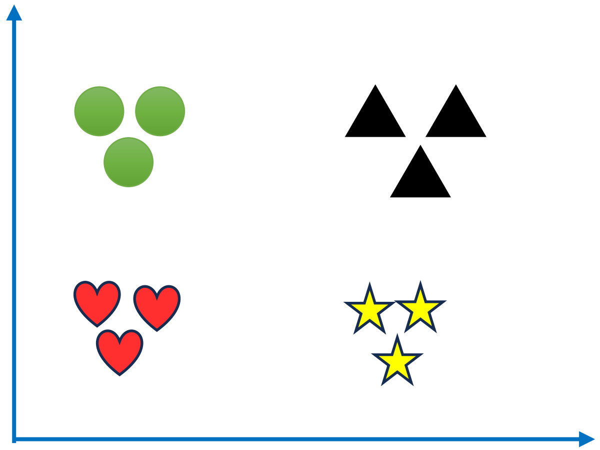


Tên	Công thức	kernel	Thiết lập hệ số
linear	$\mathbf{x}^T\mathbf{z}$	'linear'	không có hệ số
polynomial	$(r+\gamma\mathbf{x}^T\mathbf{z})^d$	'poly'	d: degree, γ: gamma, r: coef0
sigmoid	$\tanh(\gamma\mathbf{x}^T\mathbf{z}+r)$	'sigmoid'	γ: gamma, r: coef0
rbf	$\exp(-\gamma\|\mathbf{x}-\mathbf{z}\|_2^2)$	'rbf'	$\gamma > 0$: gamma

https://medium.com/geekculture/kernel-methods-in-support-vector-machines-bb9409342c49

Multiple Class Classification





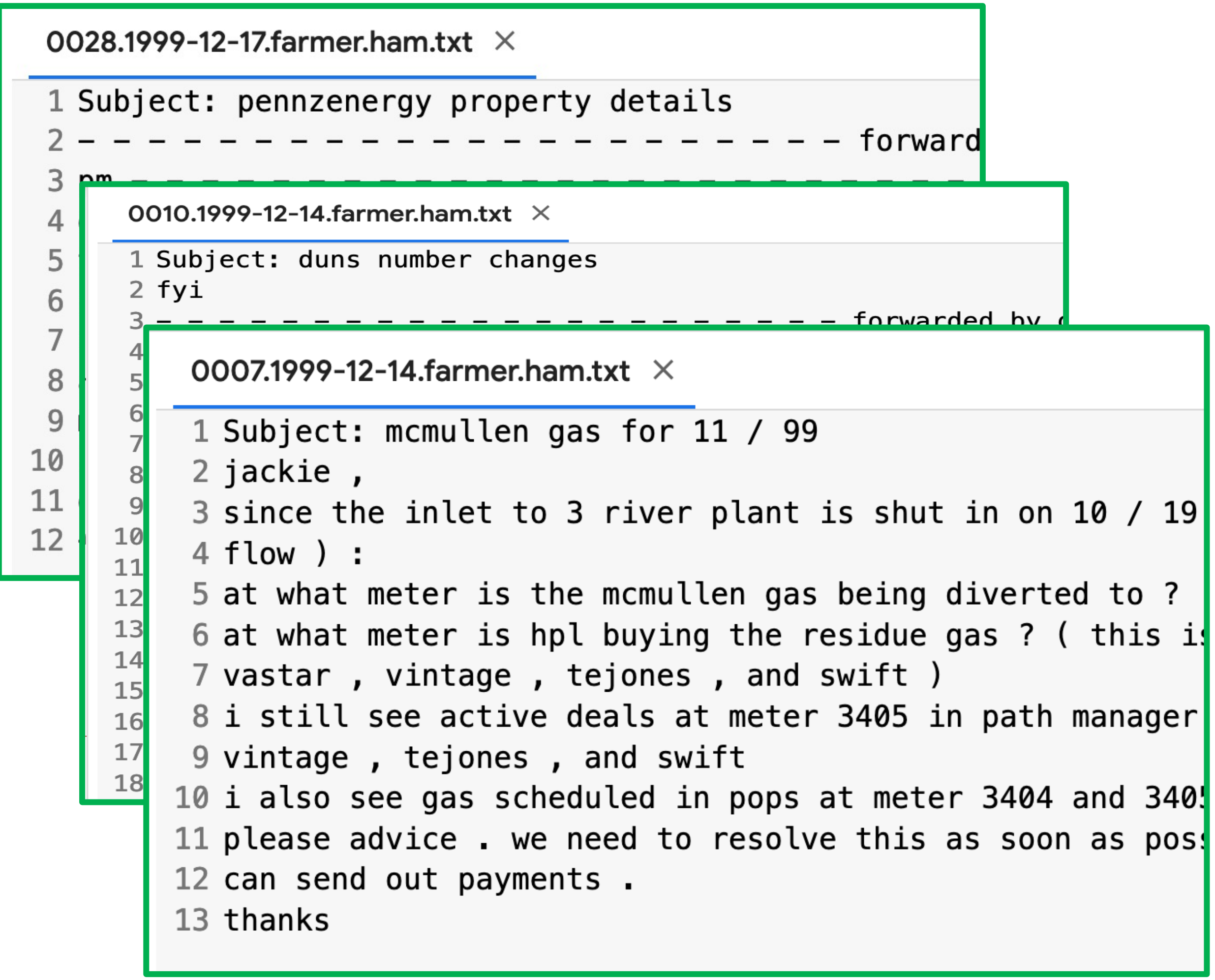
Outline

- Linear Regression to Logistic Regression
- Logistic Regression to Support Vector Machine
- Example

AI VIETNAM
All-in-One Course

Email Spam Classification - 1

Normal Email



0028.1999-12-17.farmer.ham.txt

```
Subject: pennzenergy property details
- - - - - - - - - - - - - - - - - - - - - - forward
pm - - - - - - - - - - - - - - - - - - - - - - - -
```

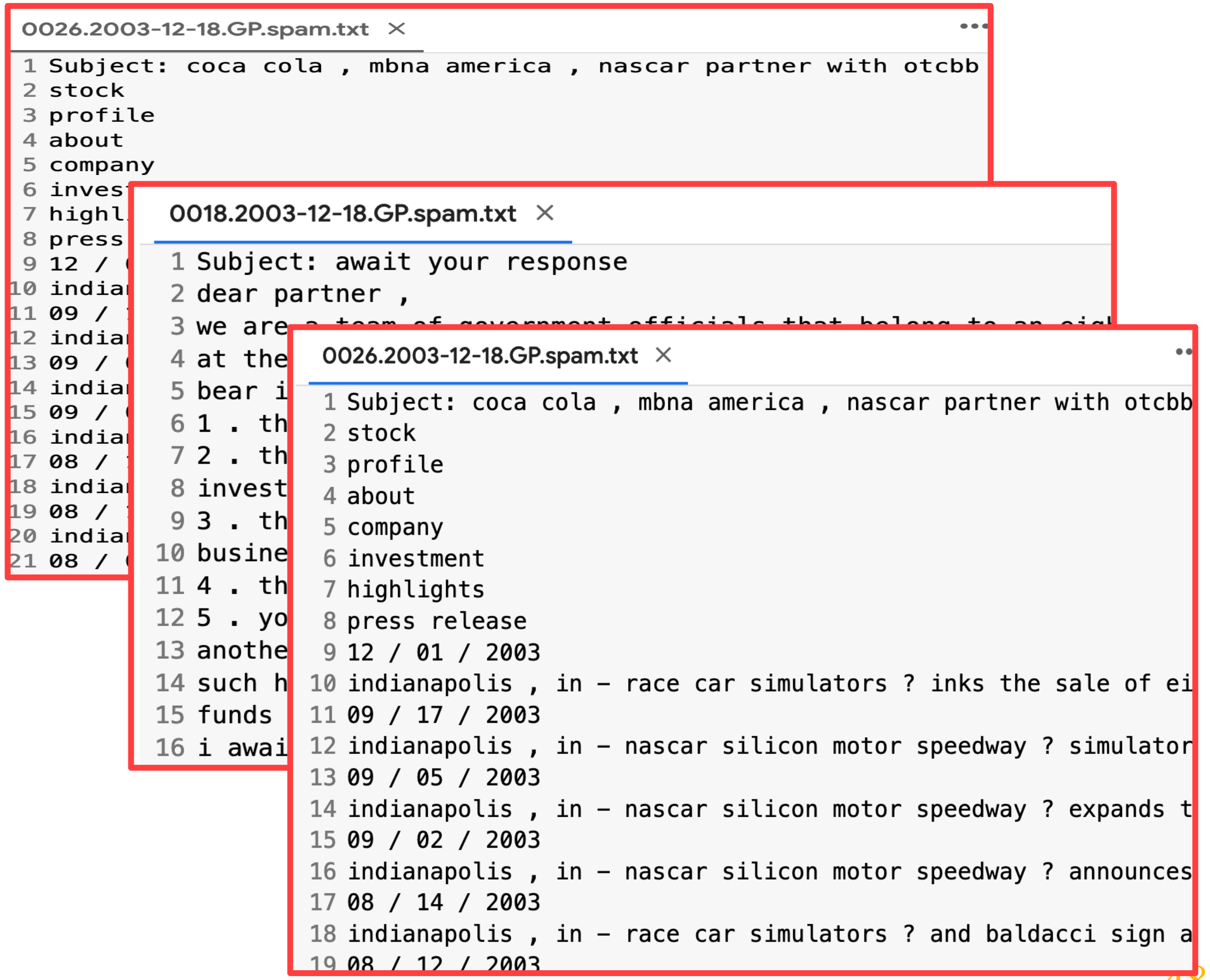
0010.1999-12-14.farmer.ham.txt

```
Subject: duns number changes
fyi
- - - - - - - - - - - - - - - - - - - - - - forwarded by
```

0007.1999-12-14.farmer.ham.txt

```
Subject: mcmullen gas for 11 / 99
jackie ,
since the inlet to 3 river plant is shut in on 10 / 19
flow ) :
at what meter is the mcmullen gas being diverted to ?
at what meter is hpl buying the residue gas ? ( this is
vastar , vintage , tejones , and swift )
i still see active deals at meter 3405 in path manager
vintage , tejones , and swift
i also see gas scheduled in pops at meter 3404 and 3405
please advice . we need to resolve this as soon as poss
can send out payments .
thanks
```

Spam Email



0026.2003-12-18.GP.spam.txt

```
Subject: coca cola , mbna america , nascar partner with otcbb
stock
profile
about
company
inves
highl
press
```

0018.2003-12-18.GP.spam.txt

```
Subject: await your response
dear partner ,
we are a team of government officials that belong to an eig
at the
bear i
1 . th
2 . th
invest
3 . th
busine
4 . th
5 . yo
anothe
such h
funds
i awai
```

0026.2003-12-18.GP.spam.txt

```
Subject: coca cola , mbna america , nascar partner with otcbb
stock
profile
about
company
investment
highlights
press release
12 / 01 / 2003
indianapolis , in - race car simulators ? inks the sale of ei
09 / 17 / 2003
indianapolis , in - nascar silicon motor speedway ? simulator
09 / 05 / 2003
indianapolis , in - nascar silicon motor speedway ? expands t
09 / 02 / 2003
indianapolis , in - nascar silicon motor speedway ? announces
08 / 14 / 2003
indianapolis , in - race car simulators ? and baldacci sign a
08 / 12 / 2003
```


Vinh Dinh Nguyen- PhD in Computer Science

Email Spam Classification - 1

Normal Email

0028.1999-12-17.farmer.ham.txt

```
Subject: pennzenergy property details
- - - - - - - - - - - - - - - - - - - - - - forward
pm - - - - - - - - - - - - - - - - - - - - - - - - -
dscottl @ . com on 12 / 14 / 99 10 : 56 : 01 am
to : ami chokshi / corp / enron @ enron
cc :
subject : pennzenergy property details
ami , attached is some more details on the devon so
me
know if you have any questions .
david
- devon stx . xls
```

Spam Email

0026.2003-12-18.GP.spam.txt

```
Subject: coca cola , mbna america , nascar partner with otcbb
stock
profile
about
company
investment
highlights
press release
12 / 01 / 2003
indianapolis , in - race car simulators ? inks the sale of eic
09 / 17 / 2003
indianapolis , in - nascar silicon motor speedway ? simulators
09 / 05 / 2003
indianapolis , in - nascar silicon motor speedway ? expands to
09 / 02 / 2003
indianapolis , in - nascar silicon motor speedway ? announces
08 / 14 / 2003
indianapolis , in - race car simulators ? and baldacci sign ag
08 / 12 / 2003
indianapolis , in - imts forms new subsidiary for manufacturin
08 / 07 / 2003
```

Text cleanup

Sample Code

```python
text_before = "Subject: vastar resources , inc ."
text_after = text_cleanup(text_before)
print("before cleanup: ", text_before)
print("after cleanup: ", text_after)
```

```
before cleanup:  Subject: vastar resources , inc .
after cleanup:  ['subject', 'vastar', 'resources', 'inc']
```

Email Spam Classification - 1

Normal Email

0028.1999-12-17.farmer.ham.txt

```
Subject: pennzenergy property details
- - - - - - - - - - - - - - - - - - - - - - forward
pm - - - - - - - - - - - - - - - - - - - - - - - - -
dscottl @ . com on 12 / 14 / 99 10 : 56 : 01 am
to : ami chokshi / corp / enron @ enron
cc :
subject : pennzenergy property details
ami , attached is some more details on the devon so
me
know if you have any questions .
david
- devon stx . xls
```

Spam Email

0026.2003-12-18.GP.spam.txt

```
Subject: coca cola , mbna america , nascar partner with otcbb
stock
profile
about
company
investment
highlights
press release
12 / 01 / 2003
indianapolis , in - race car simulators ? inks the sale of ei
09 / 17 / 2003
indianapolis , in - nascar silicon motor speedway ? simulators
09 / 05 / 2003
indianapolis , in - nascar silicon motor speedway ? expands to
09 / 02 / 2003
indianapolis , in - nascar silicon motor speedway ? announces
08 / 14 / 2003
indianapolis , in - race car simulators ? and baldacci sign ag
08 / 12 / 2003
indianapolis , in - imts forms new subsidiary for manufacturin
08 / 07 / 2003
```

Text cleanup → Build dictionary

```python
def prepare_dictionary():
  start_time = time.time()

  lmtzr = WordNetLemmatizer()
  k=0
  count = {}

  directory_in_str = "/content/email"
  directory = os.fsencode(directory_in_str)

  for file in os.listdir(directory):
      file = file.decode("utf-8")
      file_name = str(os.getcwd()) + '/email/'
      file_name = file_name + file
```

word	count
ect	489
subject	294
hou	256
enron	237
deal	119
meter	114
gas	111
com	107
please	105

Email Spam Classification - 1

Normal Email

0028.1999-12-17.farmer.ham.txt

```
Subject: pennzenergy property details
- - - - - - - - - - - - - - - - - - - - - - forward
pm - - - - - - - - - - - - - - - - - - - - - - - - -
dscottl @ . com on 12 / 14 / 99 10 : 56 : 01 am
to : ami chokshi / corp / enron @ enron
cc :
subject : pennzenergy property details
ami , attached is some more details on the devon so
me
know if you have any questions .
david
- devon stx . xls
```

Spam Email

0026.2003-12-18.GP.spam.txt

```
Subject: coca cola , mbna america , nascar partner with otcbb
stock
profile
about
company
investment
highlights
press release
12 / 01 / 2003
indianapolis , in - race car simulators ? inks the sale of ei
09 / 17 / 2003
indianapolis , in - nascar silicon motor speedway ? simulators
09 / 05 / 2003
indianapolis , in - nascar silicon motor speedway ? expands to
09 / 02 / 2003
indianapolis , in - nascar silicon motor speedway ? announces
08 / 14 / 2003
indianapolis , in - race car simulators ? and baldacci sign ag
08 / 12 / 2003
indianapolis , in - imts forms new subsidiary for manufacturin
08 / 07 / 2003
```

Text cleanup

Build dictionary

Encoding data

```python
def encode_email(fileName, words):
  file_reading = open(fileName,"r",encoding='utf-8', errors='ignore')
  words_list_array = np.zeros(words.size)
  for word in file_reading.read().split():
      word = lmtzr.lemmatize(word.lower())
      if(word in stopwords.words('english') or word in string.punctuation or l
          continue
      for i in range(words.size):
          if(words[i]==word):
              words_list_array[i] = words_list_array[i]+1
              break
  return words_list_array
```

```python
normal_sample = encode_email('/content/email/0028.1999-12-17.farmer.ham.txt', words)
print(normal_sample)

spam_sample = encode_email('/content/email/0334.2004-01-30.GP.spam.txt', words)
print(spam_sample)
```

```
[0. 1. 0. 3. 0. 0. 0. 1. 0.]
[0. 0. 0. 0. 0. 0. 0. 1. 1.]
```

Email Spam Classification - 1

Normal Email

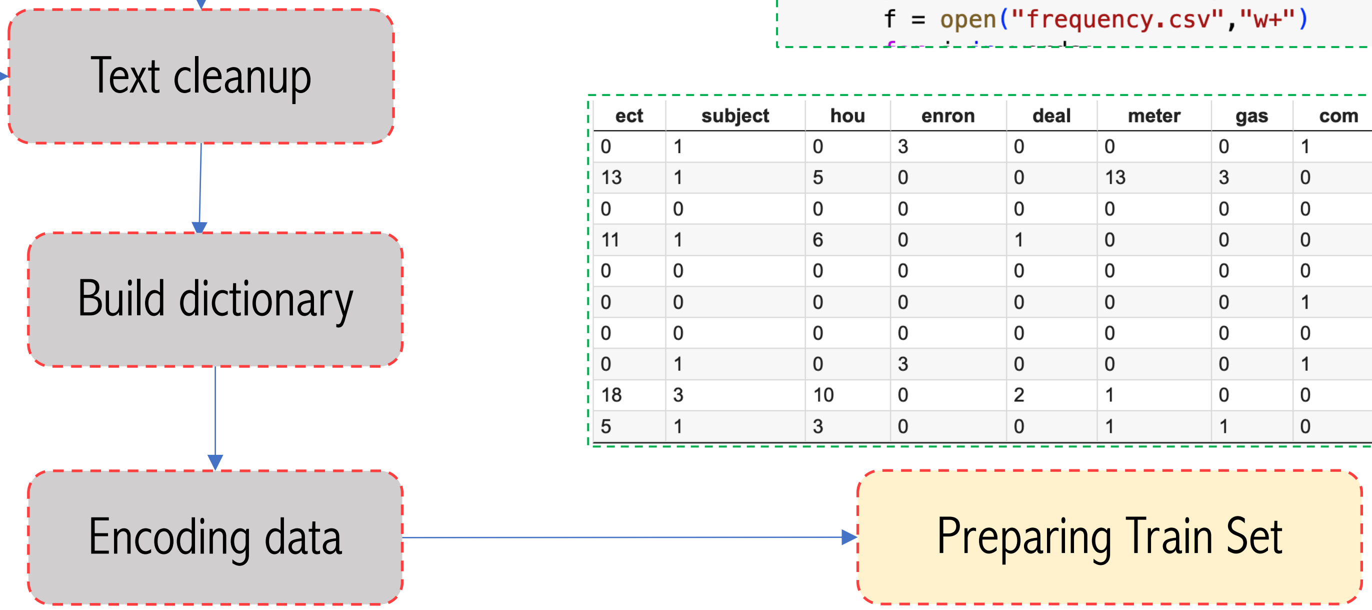
0028.1999-12-17.farmer.ham.txt

```
Subject: pennzenergy property details
- - - - - - - - - - - - - - - - - - - - - - - forward
pm - - - - - - - - - - - - - - - - - - - - - - - - -
dscottl @ . com on 12 / 14 / 99 10 : 56 : 01 am
to : ami chokshi / corp / enron @ enron
cc :
subject : pennzenergy property details
ami , attached is some more details on the devon so
me
know if you have any questions .
david
- devon stx . xls
```

Spam Email

0026.2003-12-18.GP.spam.txt

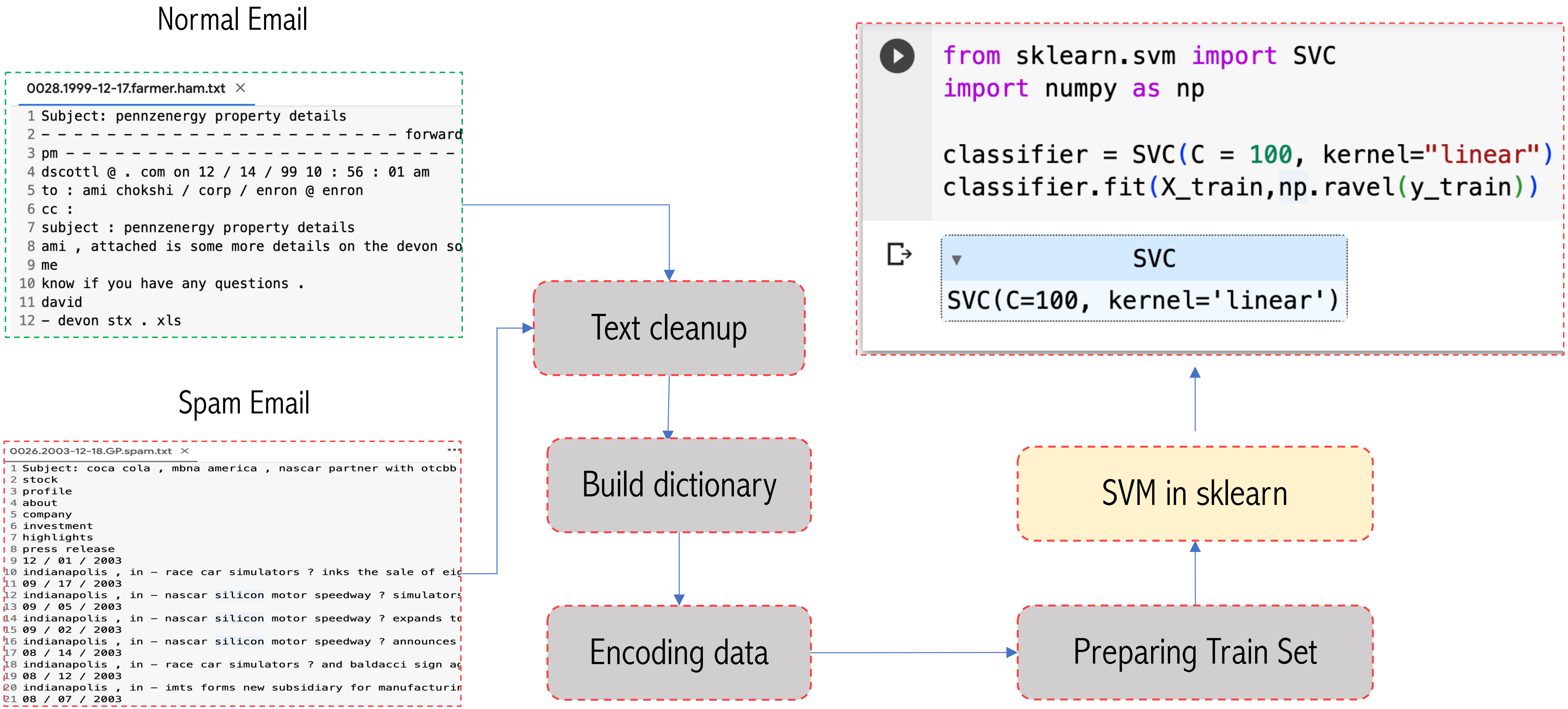
```
Subject: coca cola , mbna america , nascar partner with otcbb
stock
profile
about
company
investment
highlights
press release
12 / 01 / 2003
indianapolis , in - race car simulators ? inks the sale of eig
09 / 17 / 2003
indianapolis , in - nascar silicon motor speedway ? simulators
09 / 05 / 2003
indianapolis , in - nascar silicon motor speedway ? expands to
09 / 02 / 2003
indianapolis , in - nascar silicon motor speedway ? announces
08 / 14 / 2003
indianapolis , in - race car simulators ? and baldacci sign ag
08 / 12 / 2003
indianapolis , in - imts forms new subsidiary for manufacturin
08 / 07 / 2003
```




```python
def prepare_data():
  start_time = time()

  df = pd.read_csv('wordslist.csv',header=0)
  words = df['word']

  lmtzr = WordNetLemmatizer()

  directory_in_str = "email/"
  directory = os.fsencode(directory_in_str)
  print("****" , directory)
  f = open("frequency.csv","w+")
```

ect	subject	hou	enron	deal	meter	gas	com	please	output
0	1	0	3	0	0	0	1	0	0
13	1	5	0	0	13	3	0	2	0
0	0	0	0	0	0	0	0	0	0
11	1	6	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	1	0	3	0	0	0	1	0	0
18	3	10	0	2	1	0	0	0	0
5	1	3	0	0	1	1	0	0	0

Email Spam Classification - 1



Product Classfication - 2

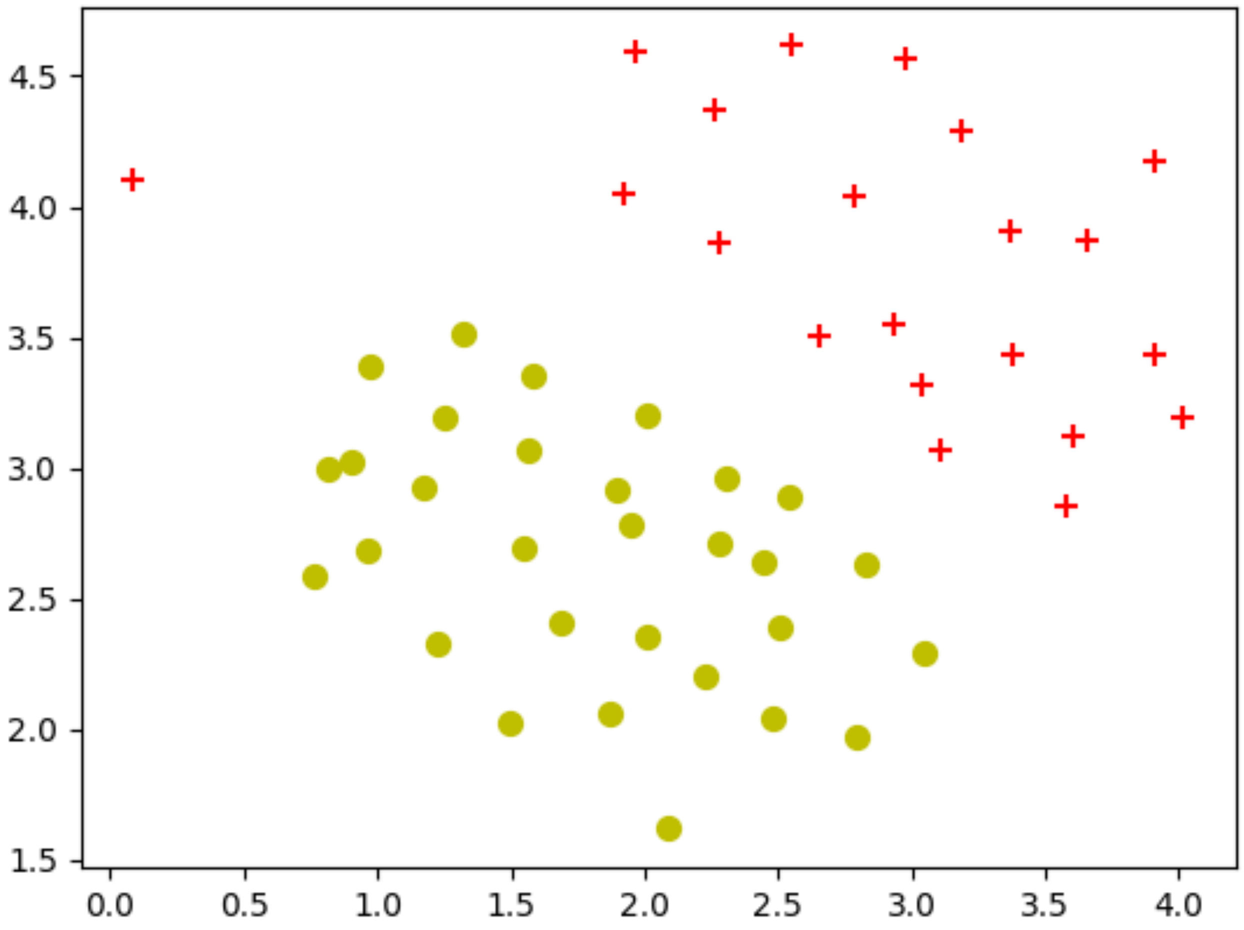
Dataset

Feature 1	Feature 2	Category
1.9643	4.5957	1
2.2753	3.8589	1
2.9781	4.5651	1
…	…	
…	….	
3.3814	3.4291	0

```
mat = loadmat("/content/product.mat")
X = mat["X"]
y = mat["y"]
```

```
m,n = X.shape[0],X.shape[1]
pos,neg= (y==1).reshape(m,1), (y==0).reshape(m,1)
plt.scatter(X[pos[:,0],0],X[pos[:,0],1],c="r",marker="+",s=50)
plt.scatter(X[neg[:,0],0],X[neg[:,0],1],c="y",marker="o",s=50)
```

Visualization



Product Classfication - 2

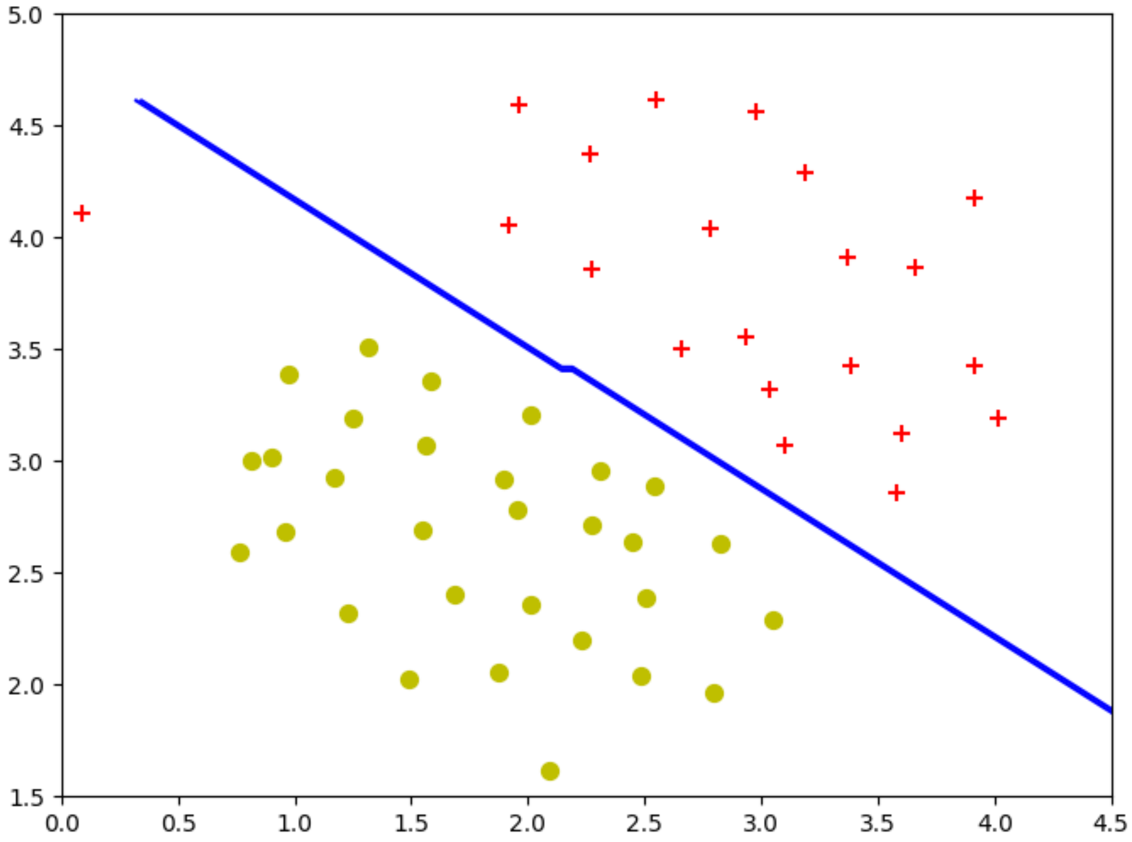
Dataset

Feature 1	Feature 2	Category
1.9643	4.5957	1
2.2753	3.8589	1
2.9781	4.5651	1
…	…	
…	….	
3.3814	3.4291	0

```python
from sklearn.svm import SVC
classifier = SVC(kernel="linear")
classifier.fit(X,np.ravel(y))
```

```
SVC
SVC(kernel='linear')
```

Visualization



Product Classfication - 2

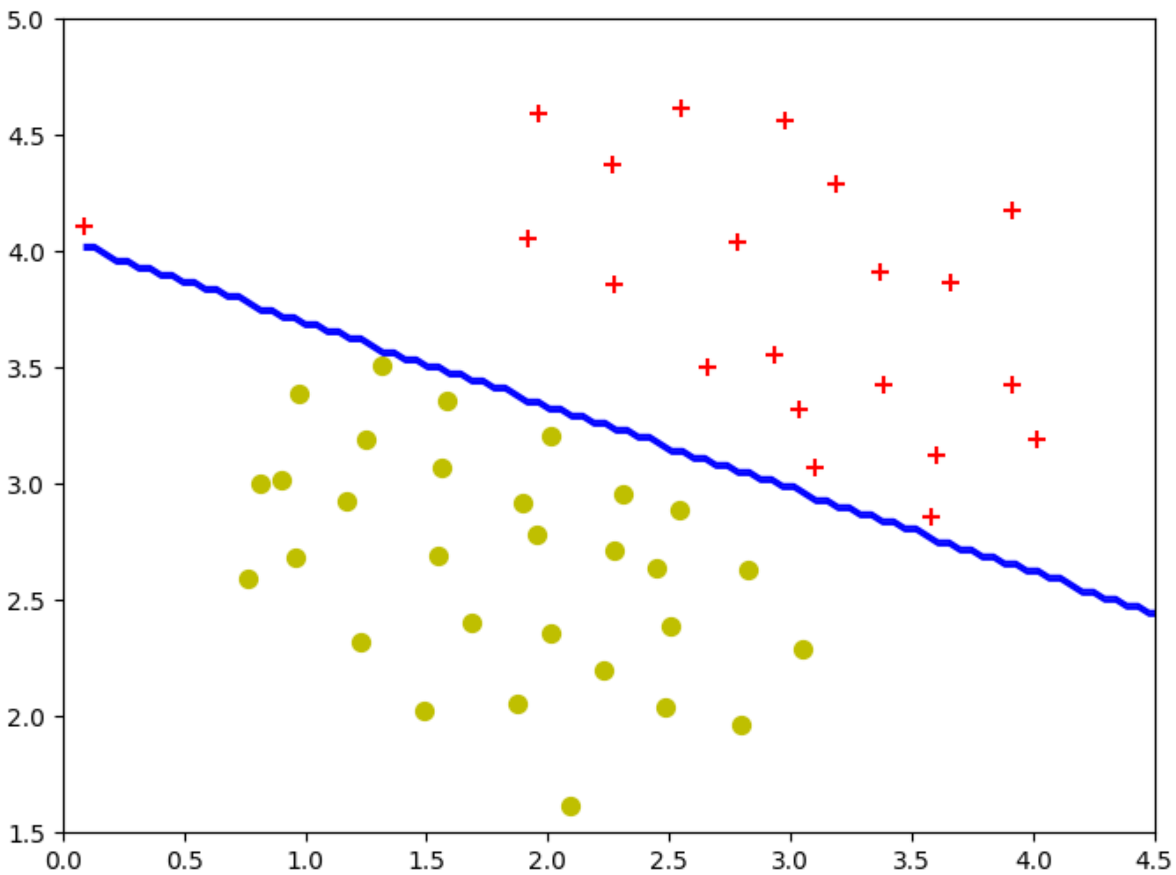
Dataset

Feature 1	Feature 2	Category
1.9643	4.5957	1
2.2753	3.8589	1
2.9781	4.5651	1
…	…	
…	….	
3.3814	3.4291	0

```
# Test C = 100
classifier_c_100 = SVC(C=100,kernel="linear")
classifier_c_100.fit(X,np.ravel(y))
```

```
SVC
SVC(C=100, kernel='linear')
```

Visualization (C = 100)



Product Classfication - 2

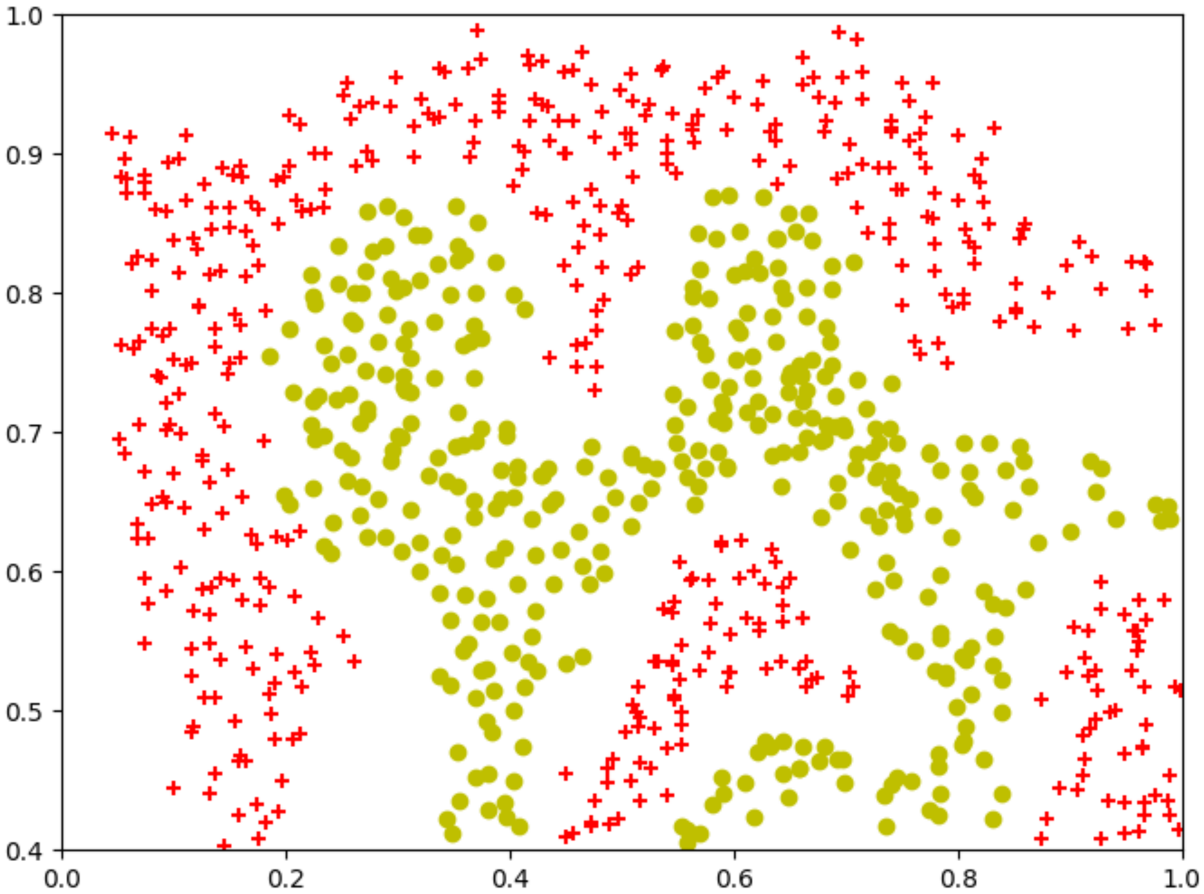
Dataset

Feature 1	Feature 2	Category
-1.58986e-01	4.23977e-01	1
-3.47926e-01	4.70760e-01	1
-5.04608e-01	3.53801e-01	1
…	…	
…	….	
-5.96774e-01	-5.96774e-01	0

```
mat2 = loadmat("/content/product_c_.mat")
X2 = mat2["X"]
y2 = mat2["y"]
```

```
m2,n2 = X2.shape[0],X2.shape[1]
pos2,neg2= (y2==1).reshape(m2,1), (y2==0).reshape(m2,1)
plt.figure(figsize=(8,6))
plt.scatter(X2[pos2[:,0],0],X2[pos2[:,0],1],c="r",marker="+")
plt.scatter(X2[neg2[:,0],0],X2[neg2[:,0],1],c="y",marker="o")
plt.xlim(0,1)
plt.ylim(0.4,1)
```

Visualization



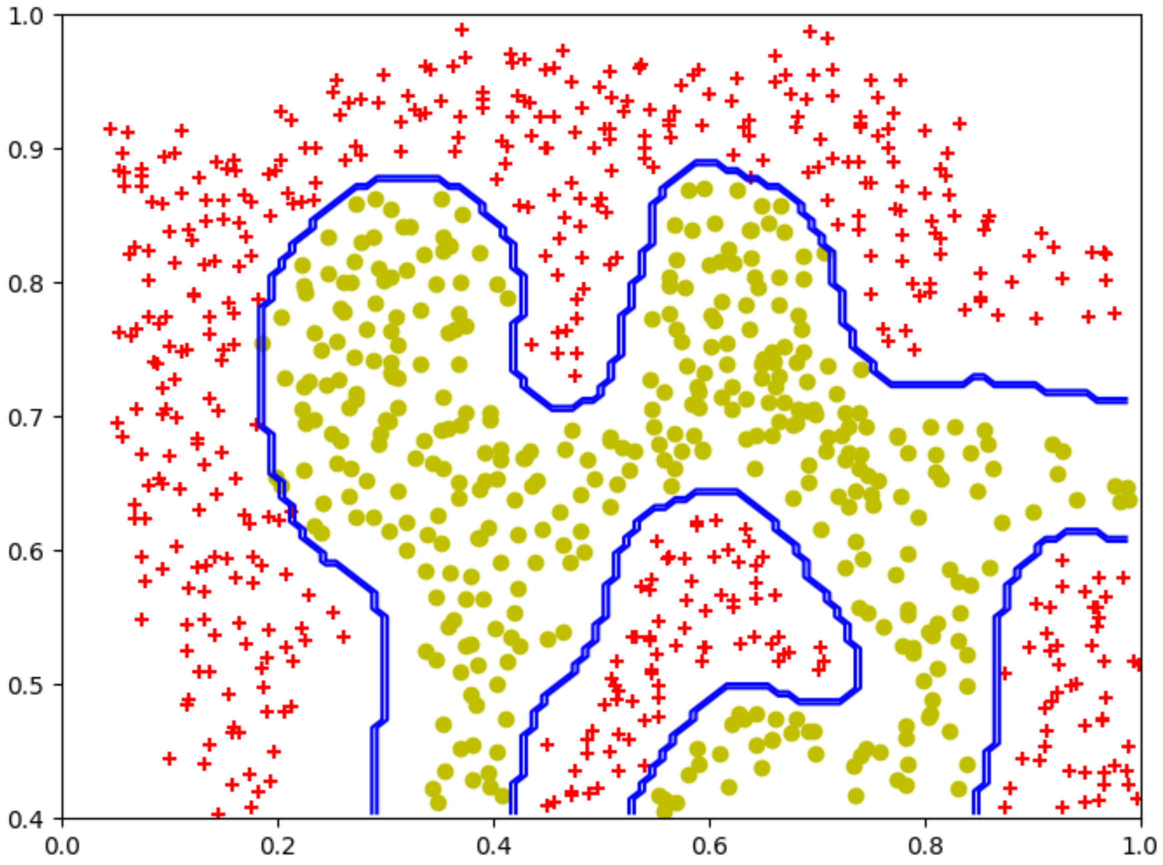
Product Classfication - 2

Dataset

Feature 1	Feature 2	Category
-1.58986e-01	4.23977e-01	1
-3.47926e-01	4.70760e-01	1
-5.04608e-01	3.53801e-01	1
…	…	
…	….	
-5.96774e-01	-5.96774e-01	0

```
from sklearn.svm import SVC
classifier = SVC(kernel="rbf", gamma=300)
classifier.fit(X2,y2.ravel())
```

Visualization




Thank you