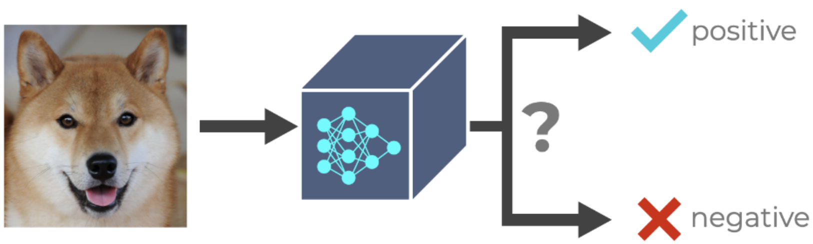
Logistic regression

Ngô Minh Nhựt

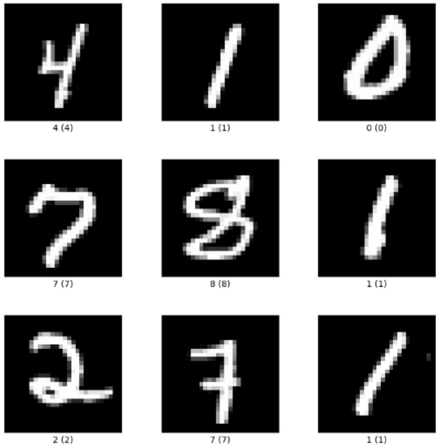
2025

Binary classification

- Answer a question with **yes** or **no**
  - Check if an email is spam
  - Check if a transaction is anormal
  - Check if a person exposes to health risk
  - Check if an area of an image contains a digit **0**



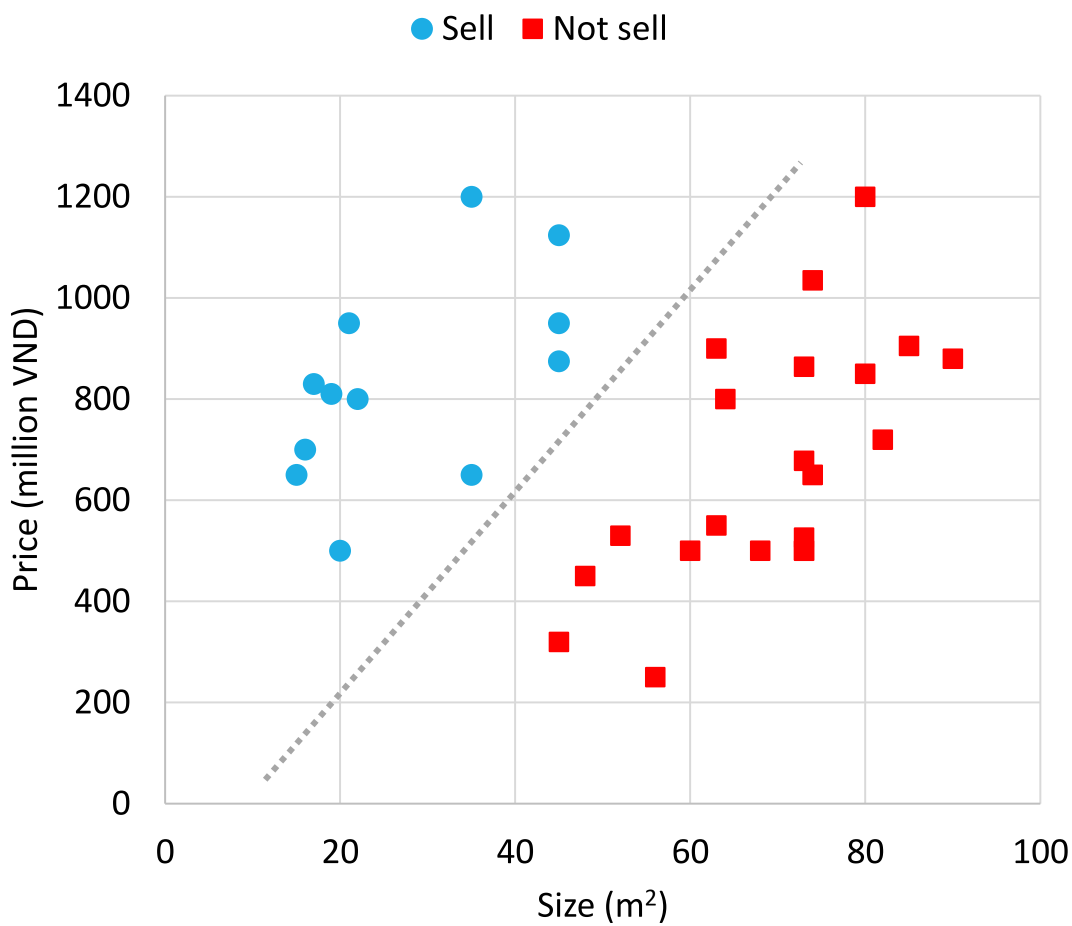
Source: Internet



MNIST dataset

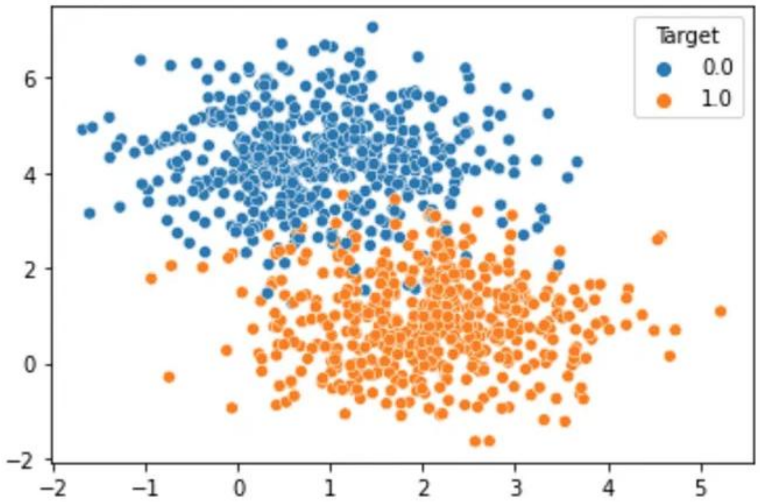
Binary classification

Size	Price	Sell?
80	600	No
30	1200	Yes
70	850	No
26	1200	No
…	…	…



Why logistic regression?

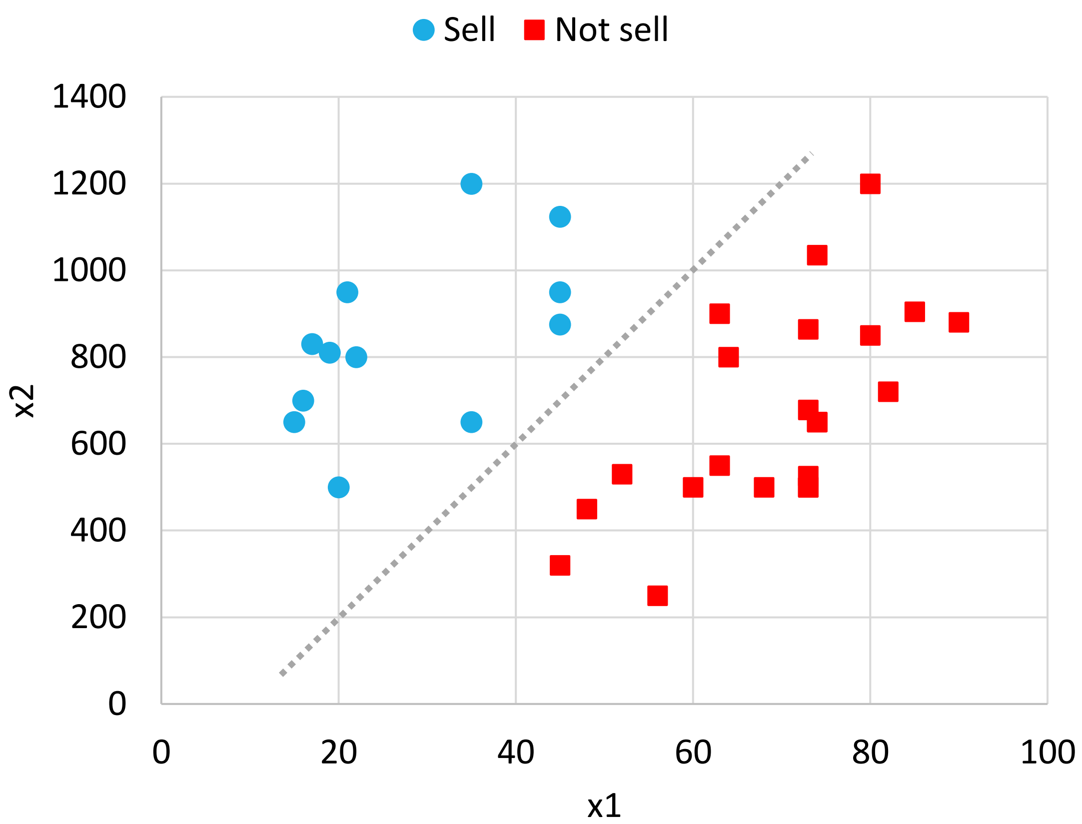
- ❑ Linear regression struggles with binary classification because it does not output probabilities.
  - Linear regression is widely used for predicting continuous outcomes.
  - In binary classification, we want outputs between 0 and 1, representing probabilities.



Source: Internet

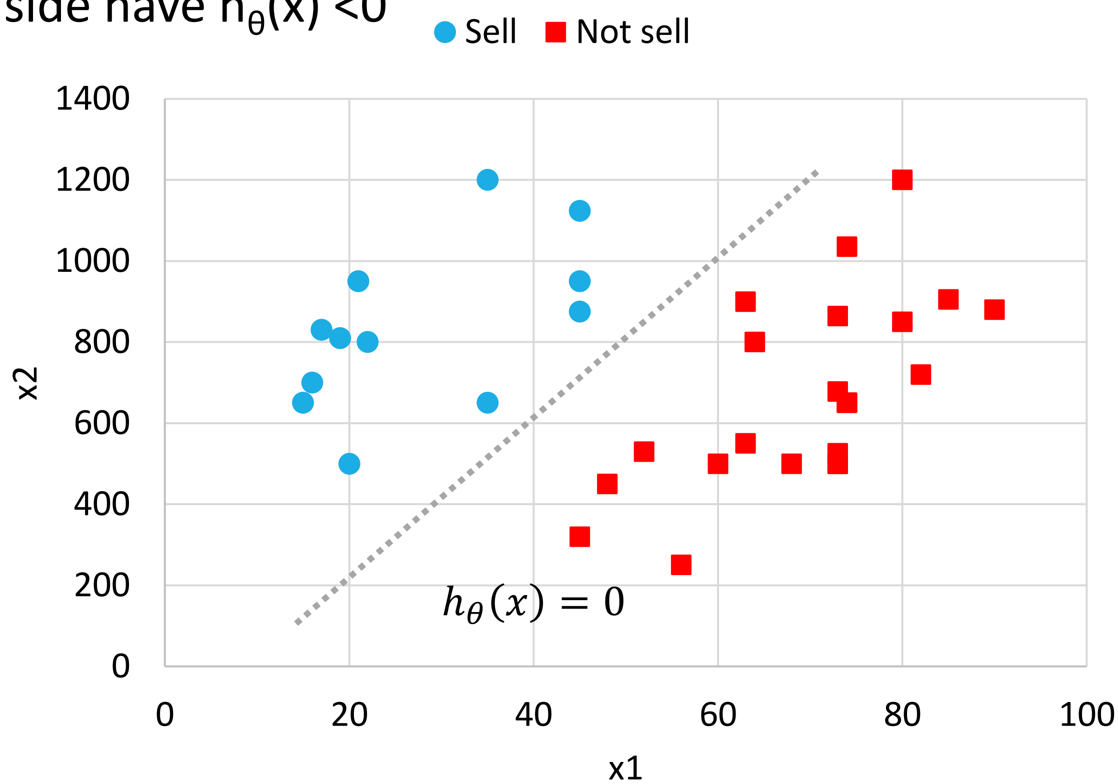
Output in number

- We need numbers for output instead of {yes, no} for calculation
- We use y = {1, 0}
  - 1: positive/yes
  - 0: negative/no



Classification boundary

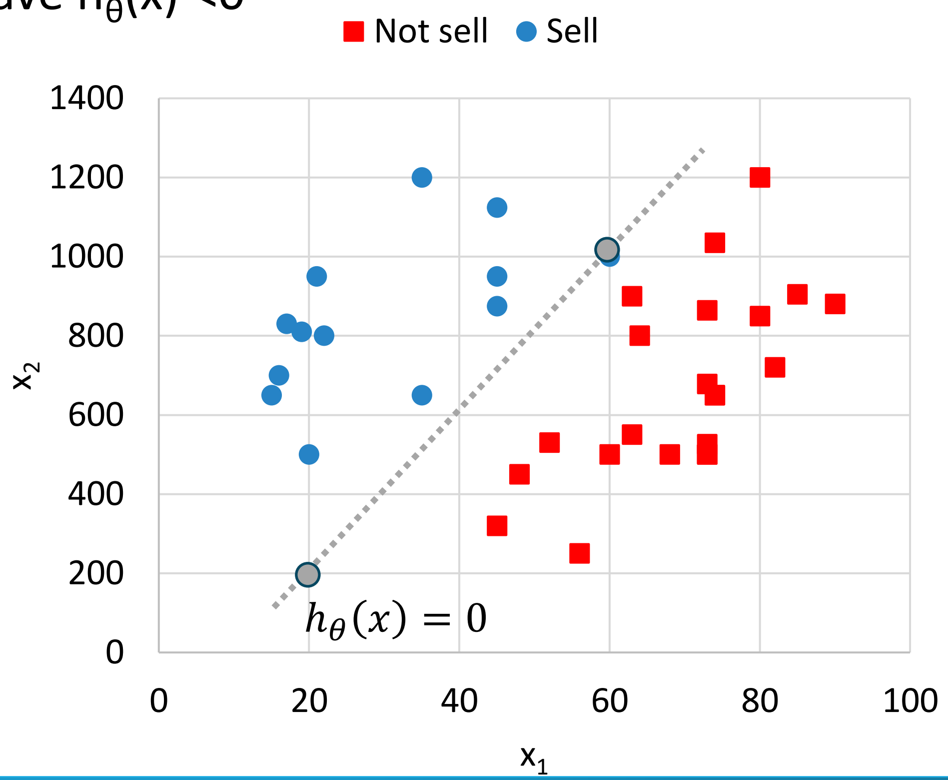
- Boundary: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$
  - Samples on the boundary have $h_\theta(x) = 0$
  - Samples on one side have $h_\theta(x) > 0$
  - Samples on the other side have $h_\theta(x) < 0$



Classification boundary

- Boundary: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$
  - **Samples on the boundary have $h_\theta(x) = 0$**
  - Samples on one side have $h_\theta(x) > 0$
  - Samples on the other side have $h_\theta(x) < 0$

- $h_\theta(x) = 200 - 20x_1 + x_2 = 0$
  - x1 = 20, x2 = 200
  - x1 = 60, x2 = 1000

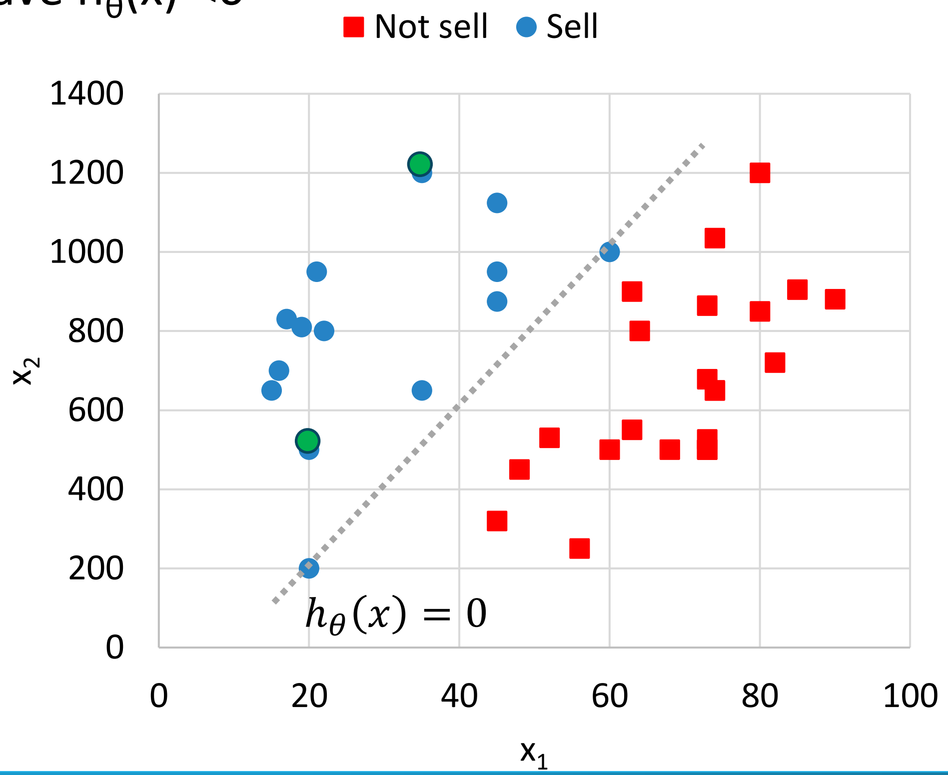


Classification boundary

- Boundary: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$
  - Samples on the boundary have $h_\theta(x) = 0$
  - **Samples on one side have $h_\theta(x) > 0$**
  - Samples on the other side have $h_\theta(x) < 0$

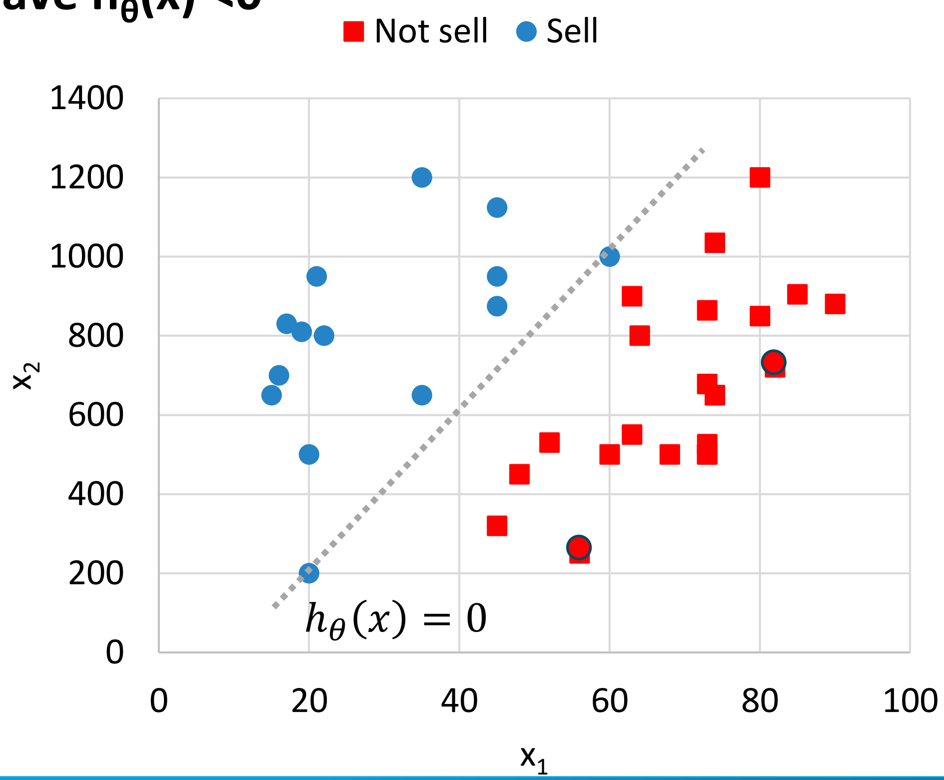
- $h_\theta(x) = 200 - 20x_1 + x_2 > 0$
  - x1 = 35, x2 = 1200
  - x1 = 20, x2 = 500

Class positive



Classification boundary

- Boundary: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$
  - Samples on the boundary have $h_\theta(x) = 0$
  - Samples on one side have $h_\theta(x) > 0$
  - **Samples on the other side have $h_\theta(x) < 0$**



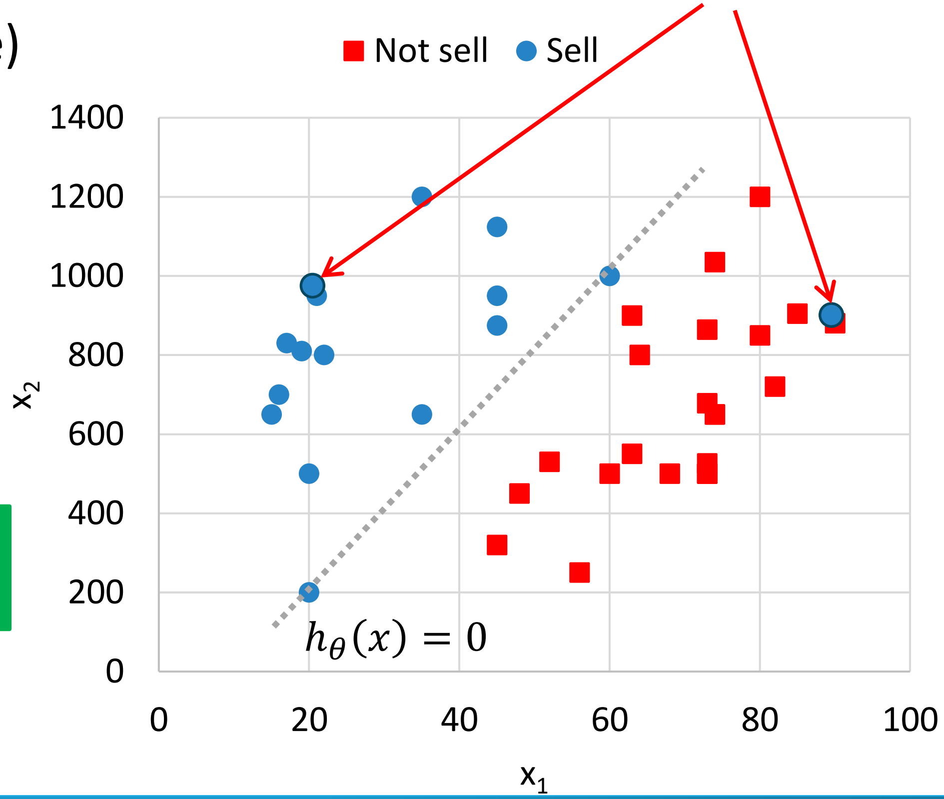
- $h_\theta(x) = 200 - 20x_1 + x_2 < 0$
  - x1 = 56, x2 = 250
  - x1 = 80, x2 = 720

Class negative

Classification boundary

- Boundary: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$
- Classification rule
  - If $h_\theta(x) \geq 0$, $\hat{y}$ = 1 (positive)
  - If $h_\theta(x) < 0$, $\hat{y}$ = 0 (negative)

Samples with high confidence



Samples are further from the boundary have more chances belonging a class

Hypothesis

- Boundary: $h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$
- Classification rule
  - If $h_\theta(x) \geq 0$, then $\hat{y}$ = 1, sample classified as positive
  - If $h_\theta(x) < 0$, then $\hat{y}$ = 0, sample classified as negative

- Output of 0 or 1 does not represent well. ✗
- A probability tells us more ✓

Model output should be how likely a sample is positive

Hypothesis

- New rule: $\begin{cases} y = 1, h_\theta(x) \to 1 \\ y = 0, h_\theta(x) \to 0 \end{cases}$

The closer the predicted is to the actual, the better the model is

y (Actual)	$h_\theta(x)$ (Predicted)
1	$\to 1$
0	$\to 0$

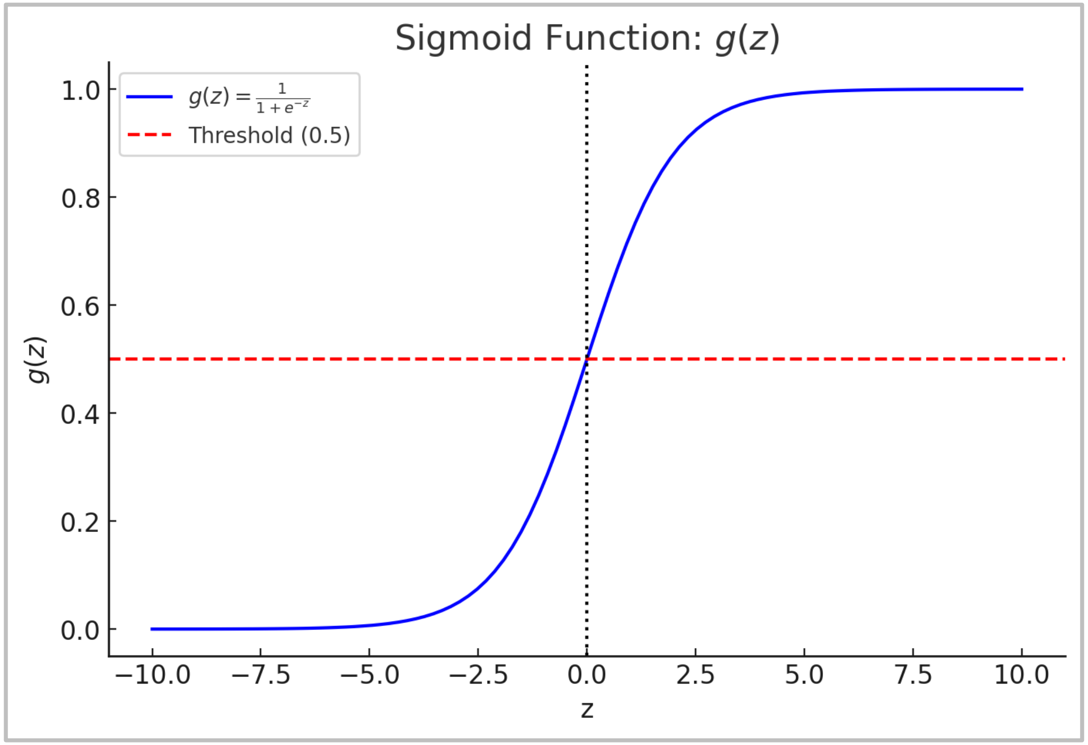
  - *Model outputs probability a sample is positive*

- New model: $h_\theta(x) = g(\theta^T x)$ with properties
  - Model output $g(\theta^T x)$ is in range (0, 1)
  - When $\theta^T x$ is much bigger than 0, $g(\theta^T x)$ approaches to 1
  - When $\theta^T x$ is much smaller than 0, $g(\theta^T x)$ approaches to 0

Hypothesis: sigmoid function

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

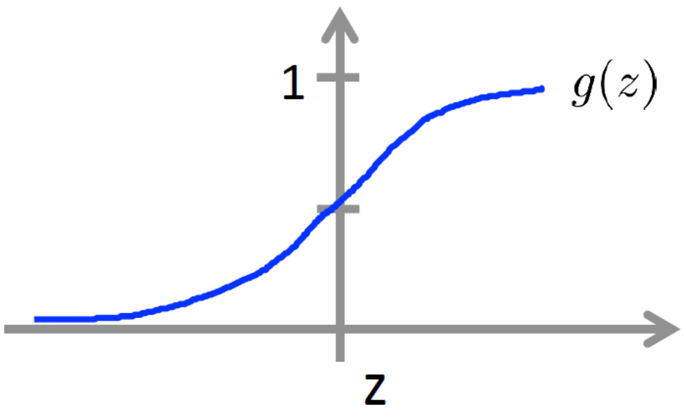
Source: ChatGPT



Sigmoid outputs values between 0 and 1, making it ideal for classification.

Hypothesis

- Model: $h_\theta(x) = g(\theta^T x)$
  - $g(z) = \frac{1}{1+e^{-z}}$
- New classification rule
  - y = 1 if $h_\theta(x) \geq 0.5$
    - Sample is classified as positive
  - y = 0 if $h_\theta(x) < 0.5$
    - Sample is classified as negative

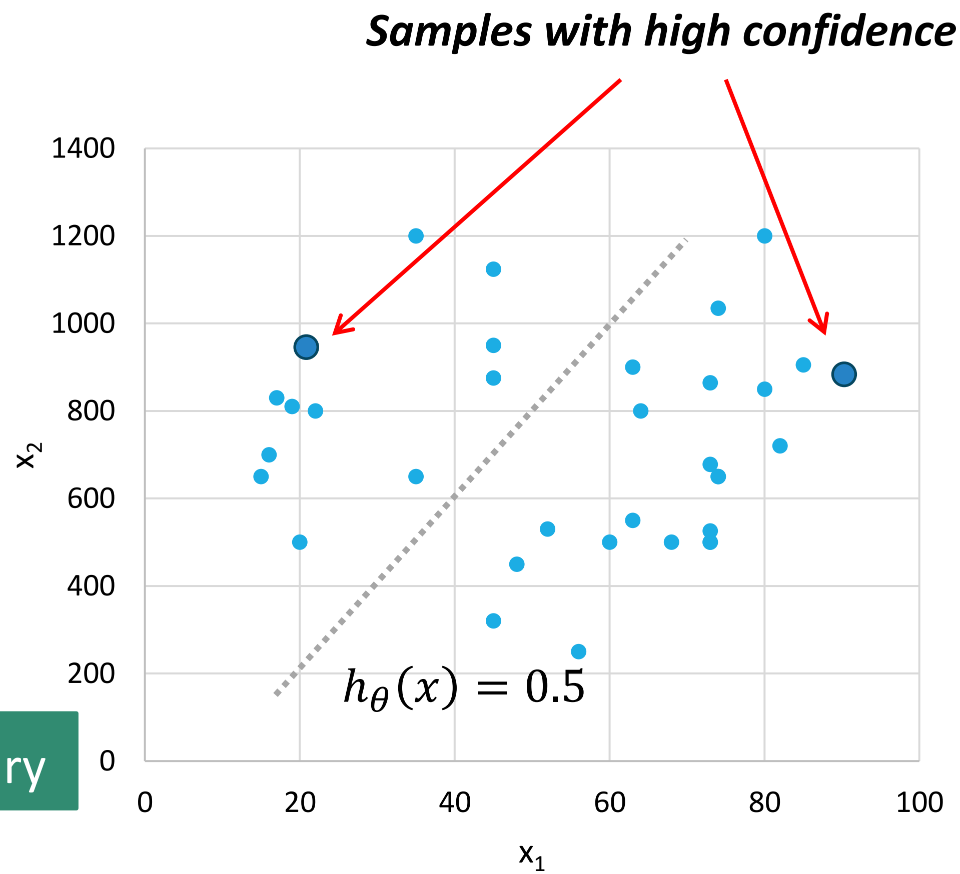


Model output shows how likely a sample is positive

Hypothesis

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

- $h_\theta(x)$ is probability y = 1
  - $h_\theta(x) = P(y = 1|x; \theta)$



$h_\theta(x) = 0.5$ is decision boundary

Cost function

- Hypothesis: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$
  - $0 \le h_\theta(x) \le 1$
- Logarithmic loss

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

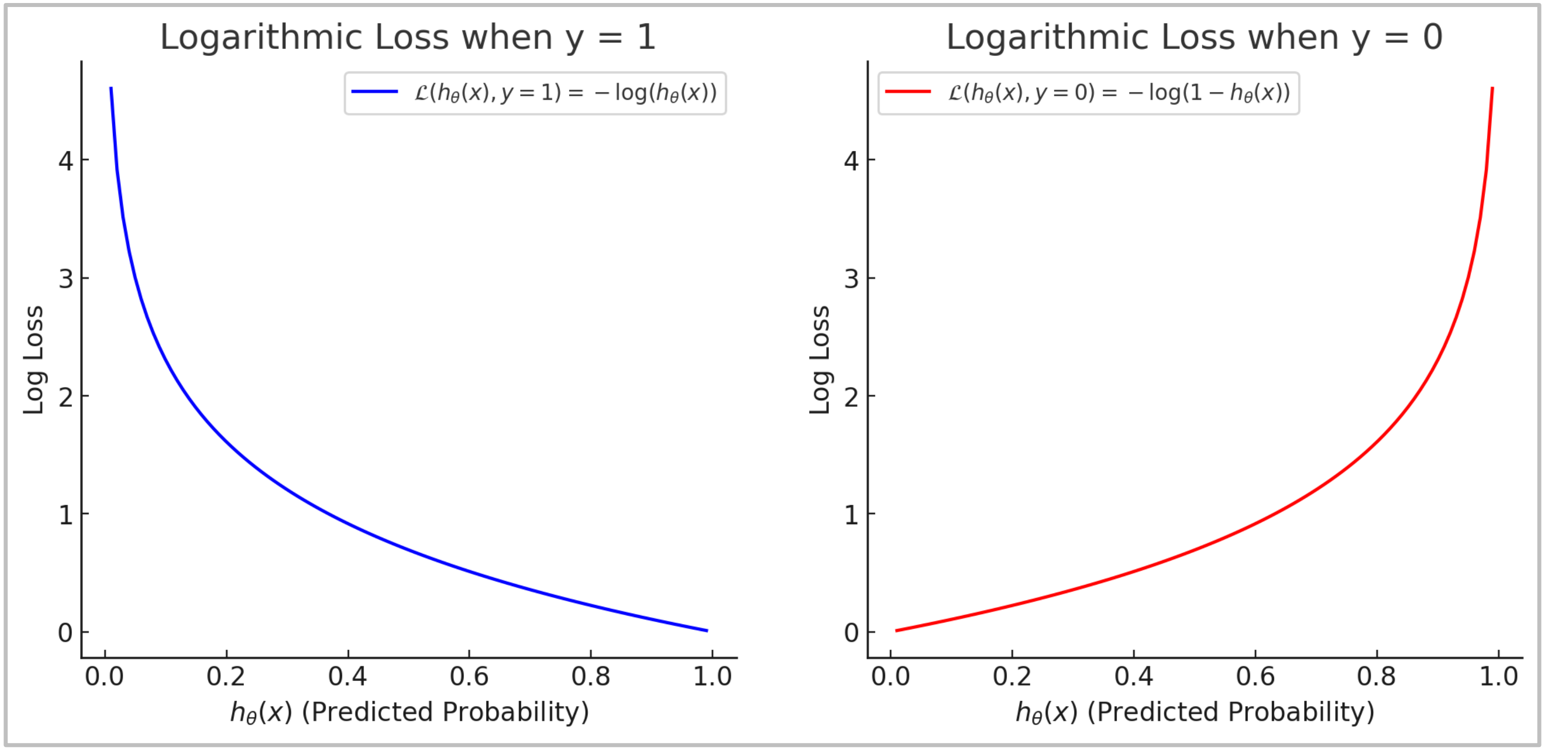
- Positive sample y = 1
  - If $h_\theta(x) \to 1$, cost $\to$ 0
  - If $h_\theta(x) \to 0$, cost $\to$ infinity
- Negative sample y = 0
  - If $h_\theta(x) \to 0$, cost $\to$ 0
  - If $h_\theta(x) \to 1$, cost $\to$ infinity

> - y: actual output
> - $h_\theta(x)$: predicted output

Cost function

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

- y: actual output
- $h_\theta(x)$: predicted output

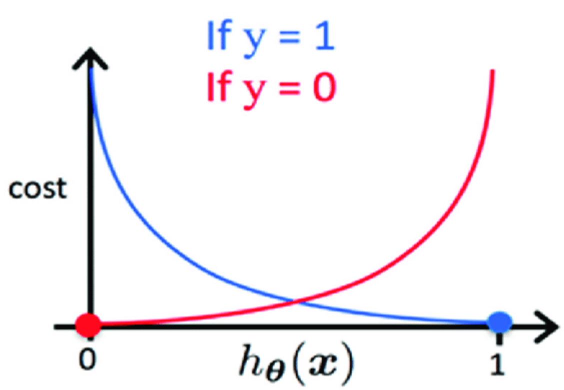


Cost function

- Hypothesis: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$
- $\text{Cost}(h(x), y) = \begin{cases} -\log h_\theta(x) & \text{if } y = 1 \\ -\log\left(1 - h_\theta(x)\right) & \text{if } y = 0 \end{cases}$
- In a new form

$$\text{Cost}(h(x), y) = -y^{(i)} \log h\left(x^{(i)}\right) - \left(1 - y^{(i)}\right) \log\left(1 - h\left(x^{(i)}\right)\right)$$

  - y = 1, 1 - y = 0 → $\text{Cost}(h(x), y) = ?$
  - y = 0, 1 - y = 1 → $\text{Cost}(h(x), y) = ?$



Source: Internet

Cost function

- Hypothesis: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$
- Const function on one sample

$$\text{Cost}(h(x), y) = -y^{(i)} \log h\left(x^{(i)}\right) - \left(1 - y^{(i)}\right) \log\left(1 - h\left(x^{(i)}\right)\right)$$

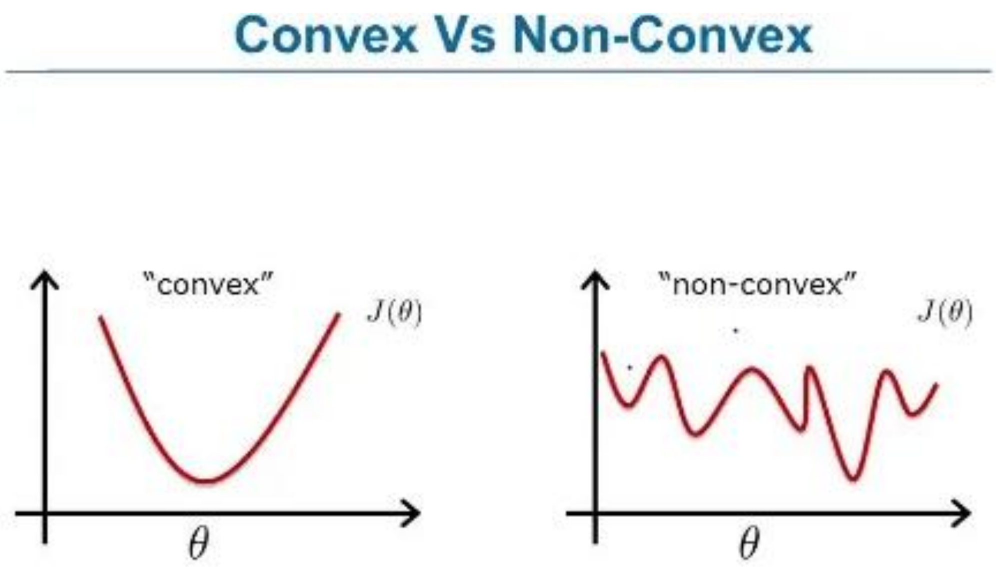
- Cost function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h\left(x^{(i)}\right)\right)$$

Model is learnt by minimizing cost function with respect to parameters

Gradient descent

- Logarithmic loss is convex, thus can be optimized with gradient descent

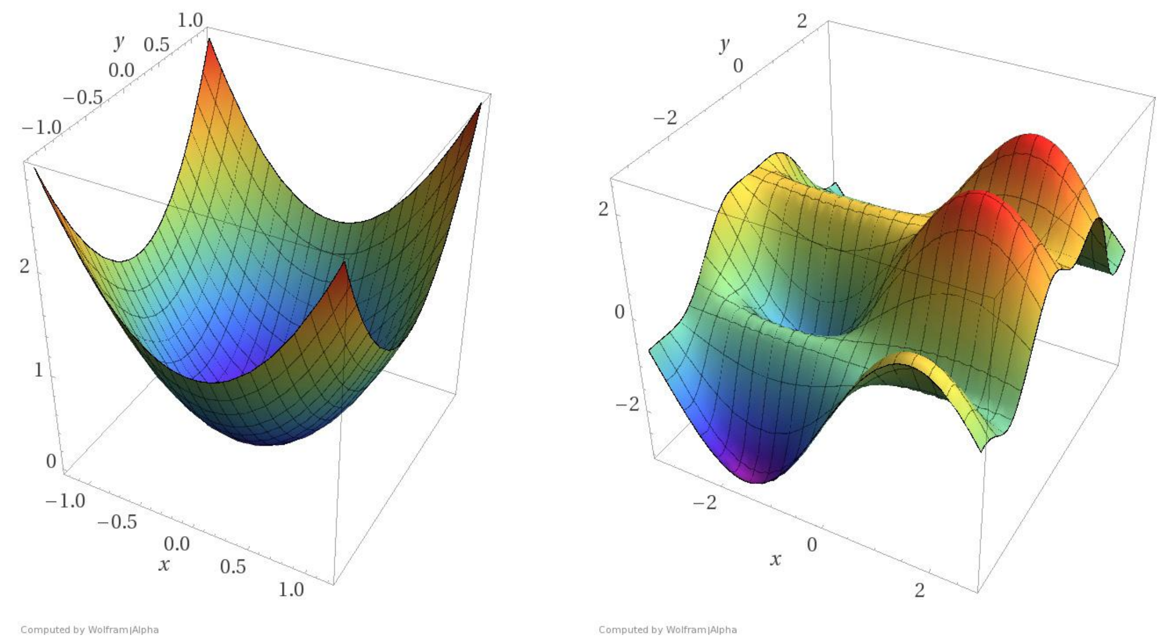


Source: Internet

Gradient descent

- Logarithmic loss is convex, thus can be optimized with gradient descent

Convex Vs Non-Convex



Source: Internet

Partial derivative

- Model: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$
- Cost: $J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$
- Partial derivative: $\frac{dJ}{d\theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$

- To calculate the derivatives
  - $g'(z) = \left(1 - g(z)\right)g(z)z'$
  - $\log(z') = \frac{z'}{z}$

See calculations at: https://ml-explained.com/blog/logistic-regression-explained

Gradient descent

- Cost function:

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

- Find θ so that $J(\theta)$ reaches minimal

Gradient descent

- Vector gradient:
  - $\frac{dJ}{d\theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$
  - $j = 0, 1, 2, \dots, n$
- Repeat until convergence

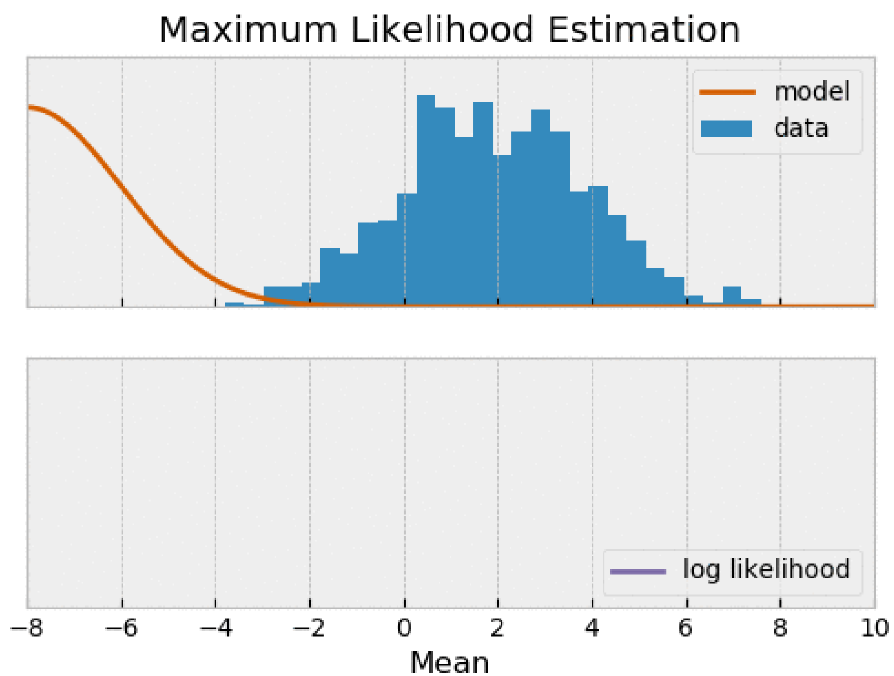
$$\{$$

$$\theta_j = \theta_j - \alpha\frac{dJ}{d\theta_j}$$

$$\}$$

Gradient descent

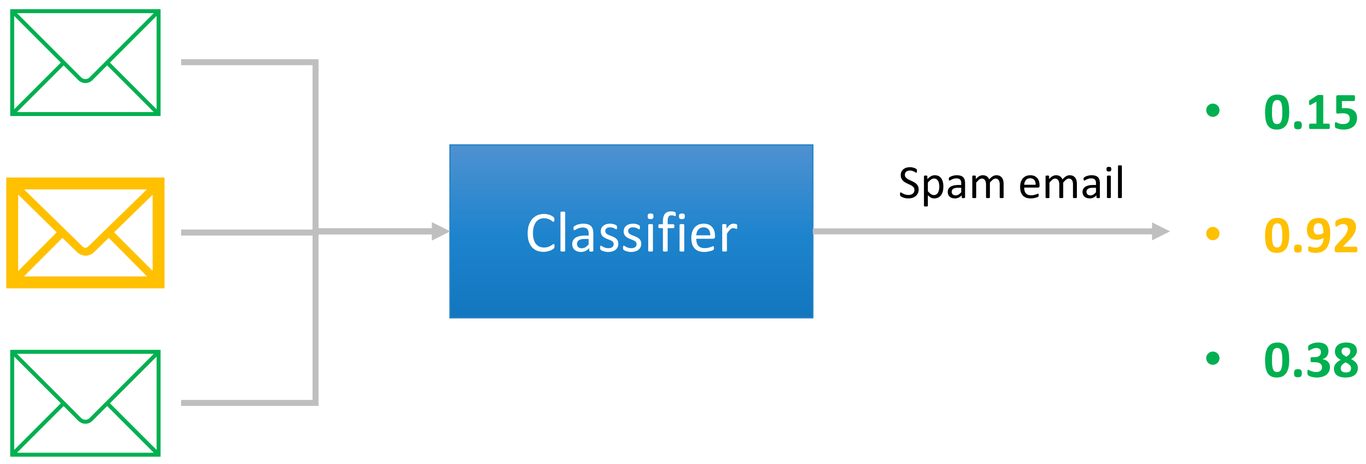
- Minimizing cost function is a maximum likelihood estimation, finding the most likely model that could have produced the observed data.



Source: Internet

Predict for a new input

- With learnt parameters θ, we can predict how likely a sample is positive
  - Output: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$



Logistic regression with multi-classes

- Weather: sunny, cloudy, rain, and heavy rain
- Digit: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Species of Iris flower: Versicolor, Setosa, and Virginica
- Object: human, cat, house, and landscape

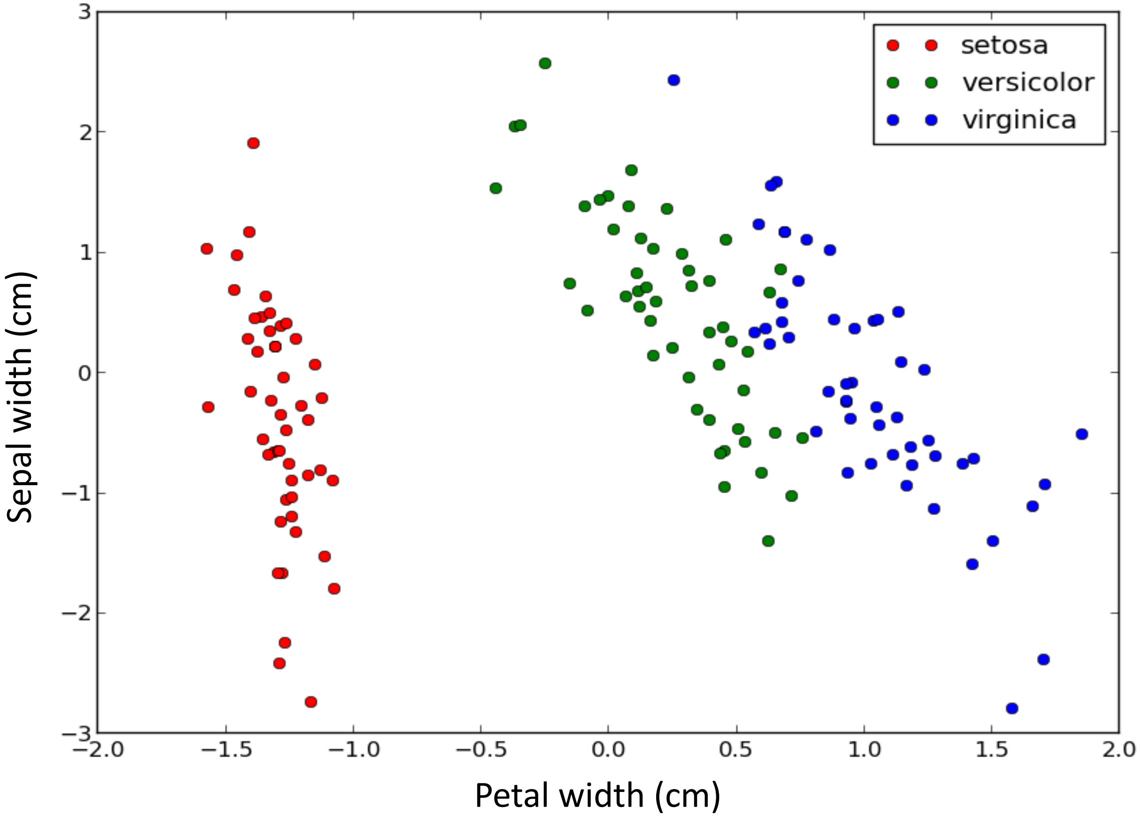
$\rightarrow$ y = {1, 2, 3 , ...}

Source: Internet

Logistic regression with multi-classes

Source: Internet

- Iris dataset

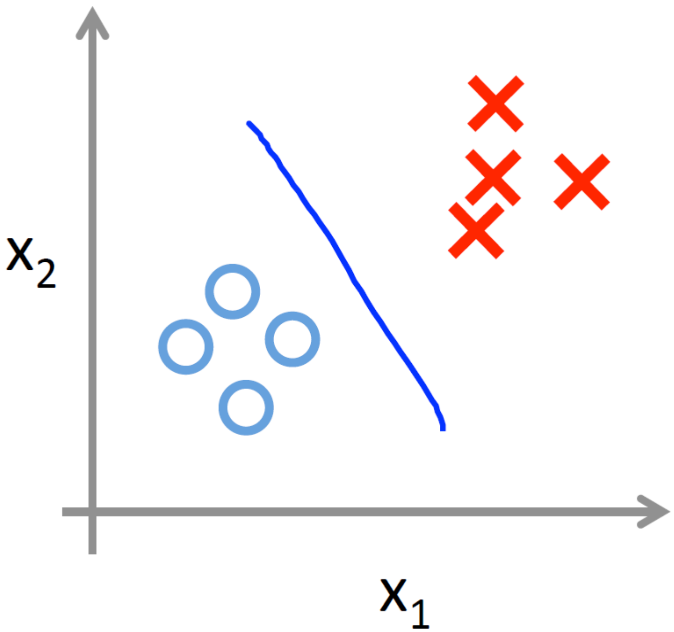


Classification of three species of Iris flower based on size of petal and sepal

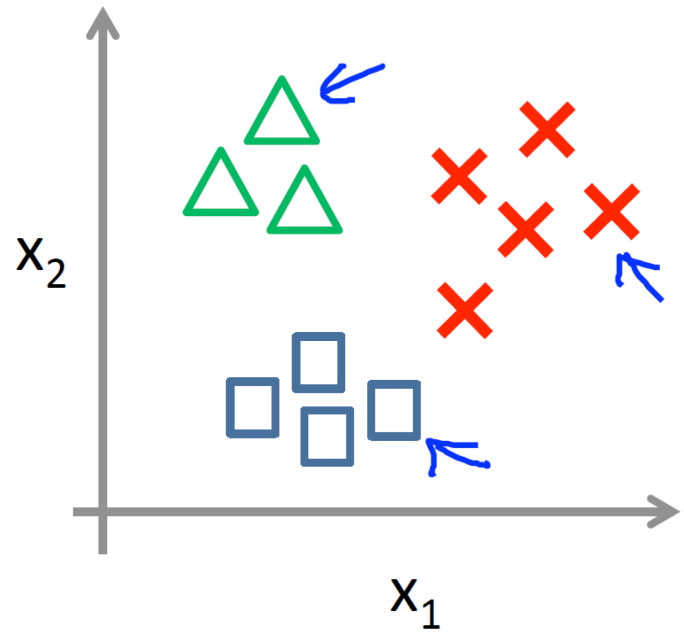
Logistic regression with multi-classes

- Binary vs Multi-class

Binary classification:



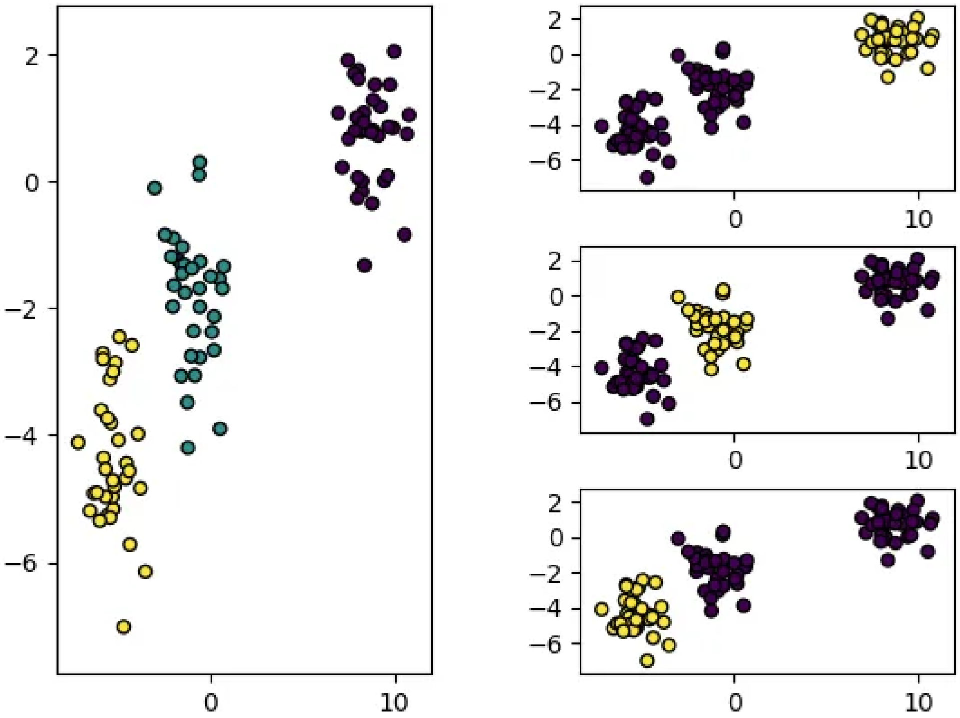
Multi-class classification:



Source: Andrew Ng

Logistic regression with multi-classes

- Logistic regression solves multi-classes problem by dividing it into n subproblems

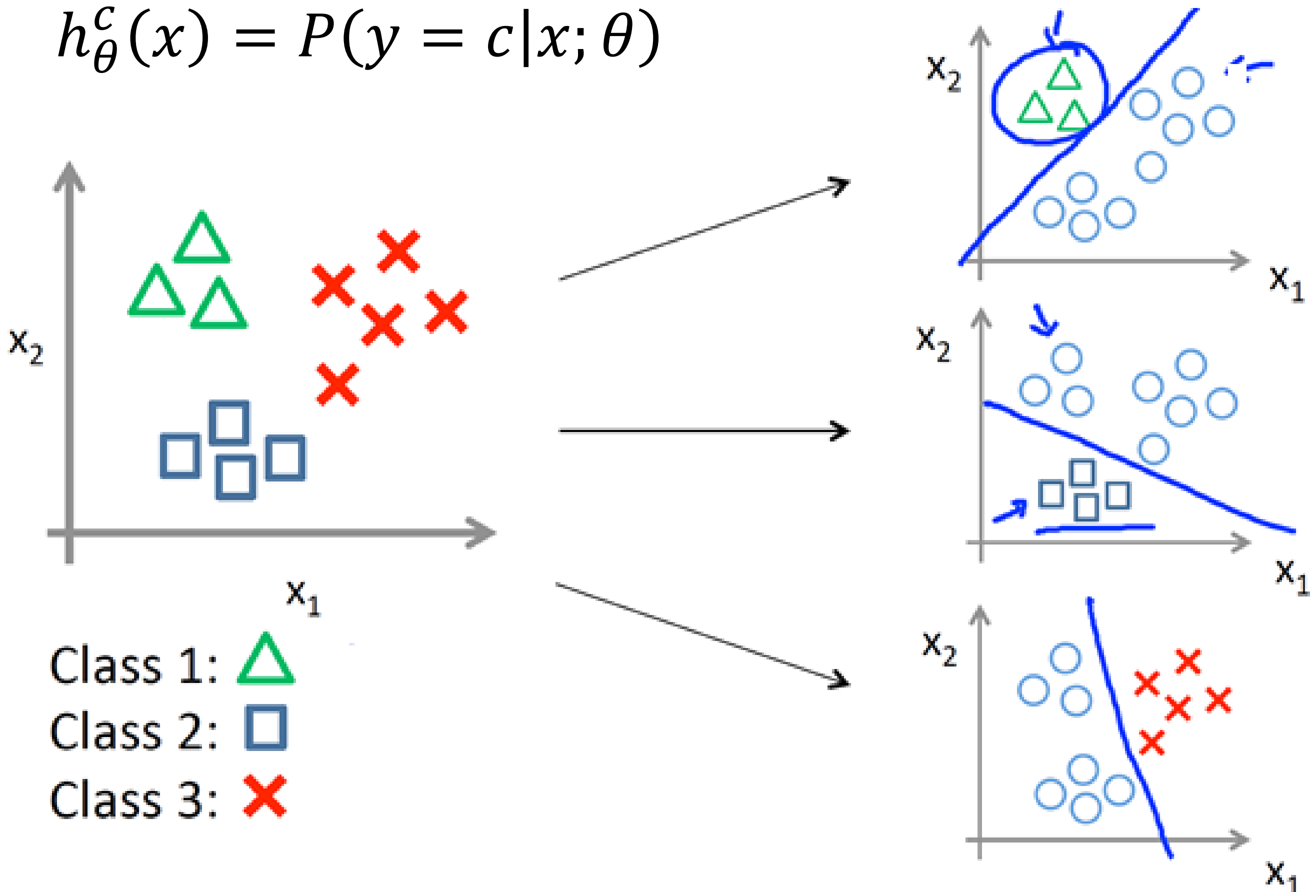


Source: Internet

Logistic regression with multi-classes

- Train classifier for each class $h_\theta^c(x)$

$$h_\theta^c(x) = P(y = c|x;\theta)$$



Source: Andrew Ng

Logistic regression with multi-classes

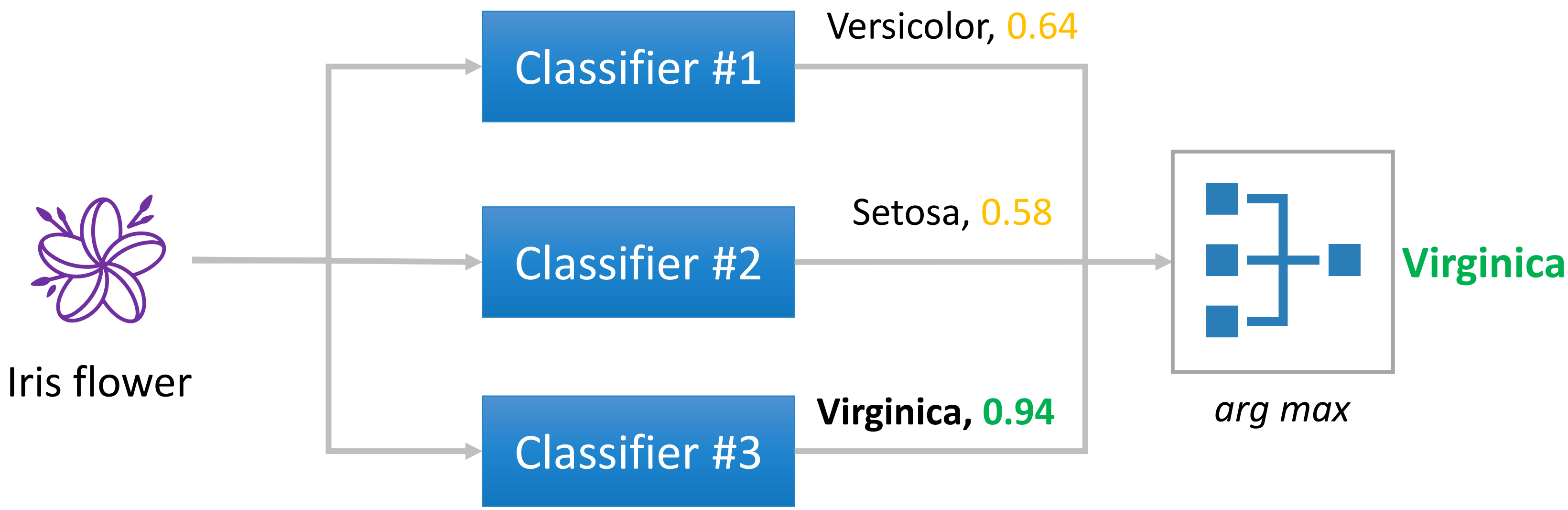
- Train classifier for each class $h_\theta^c(x)$

$$h_\theta^c(x) = P(y = c|x; \theta)$$

- Predict for a new input
  - $y = \max_c h_\theta^c(x)$
  - Select model giving maximum output

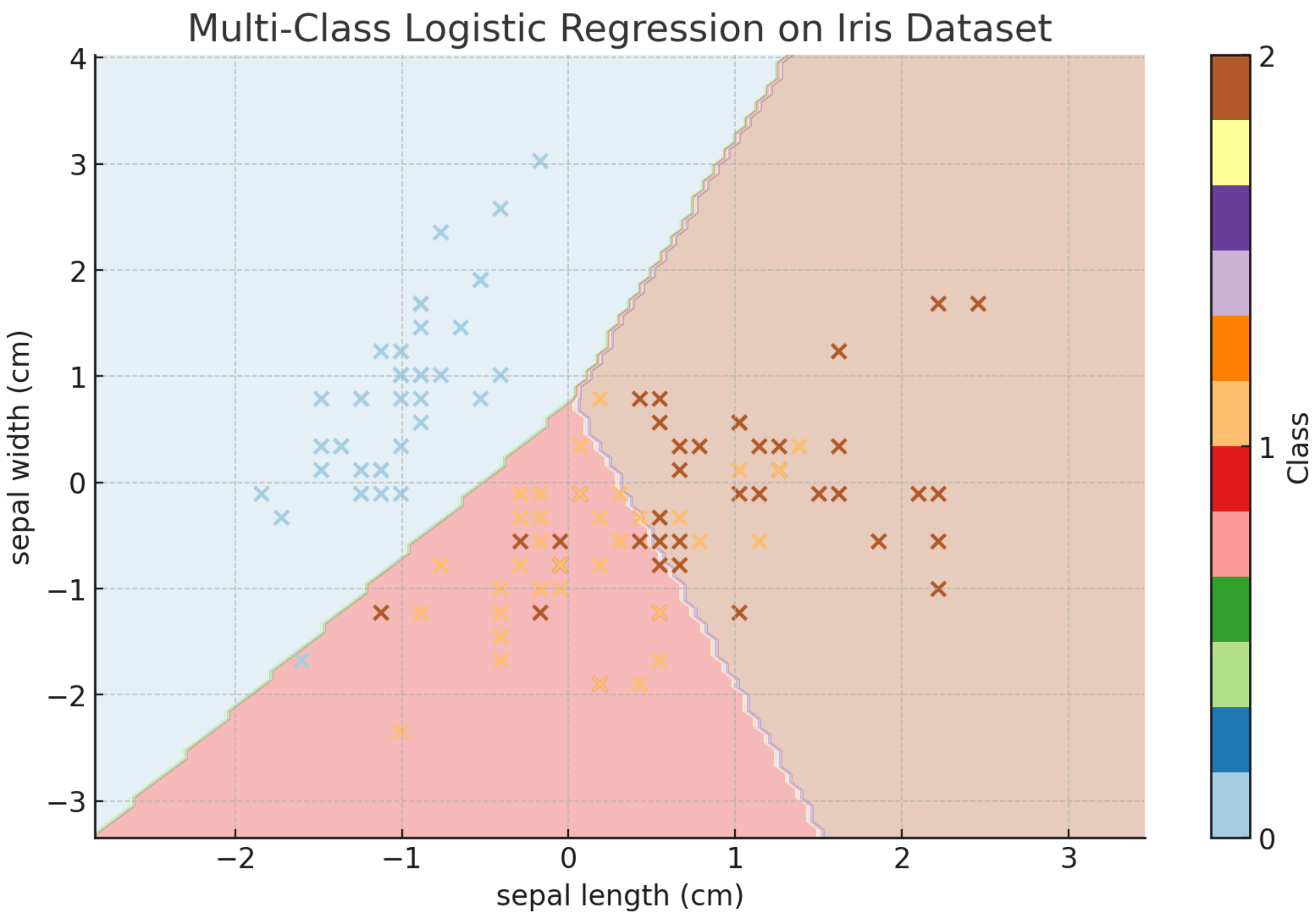
Logistic regression with multi-classes

- Select model giving maximum output

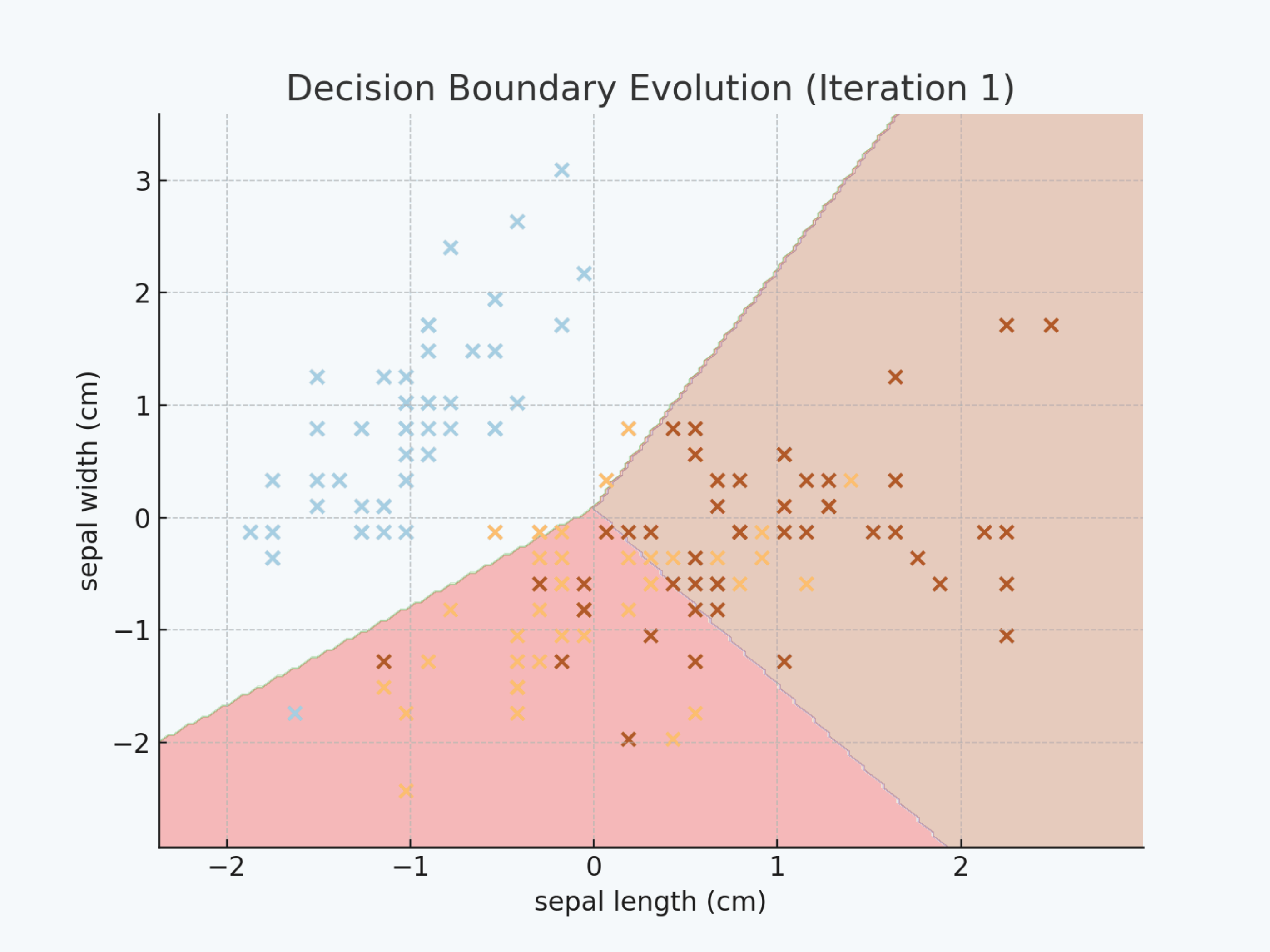


Multi-Class Logistic Regression on Iris Dataset

Logistic regression with multi-classes



Logistic regression with multi-classes



Logistic regression with multi-classes