CADT
Cambodia Academy of Digital Technology

Logistic Regression

Lecturer: Soklay HENG

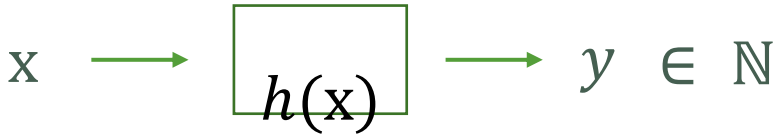
Academic Year: 2026-2027

Outline

1. Introduction
2. Hypothesis
3. Decision Boundary
4. Cost Function
5. Gradient Descent
6. Multi-Class Classification
7. Pros & Cons

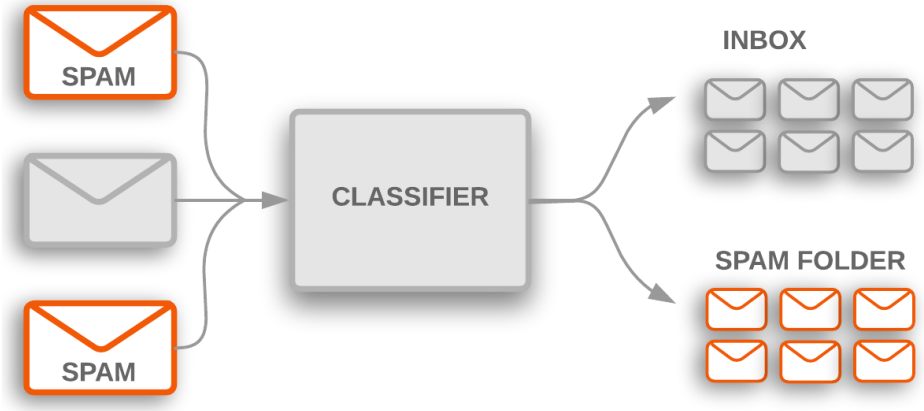
1. Introduction

Classification is a problem of supervised learning, which tries to learn a mapping function from any input variables to output variable, which is a **discrete** or **categorical value**.



1. Introduction

Examples:



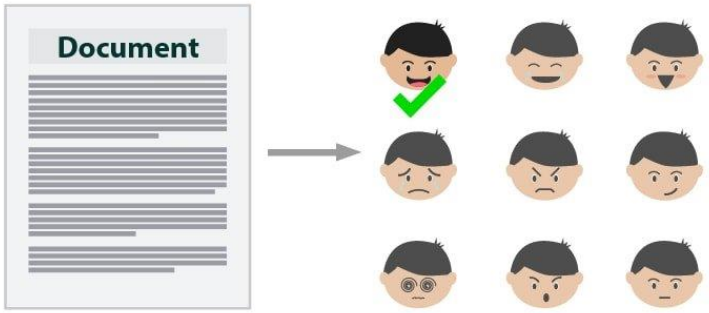
Email Spam Detection



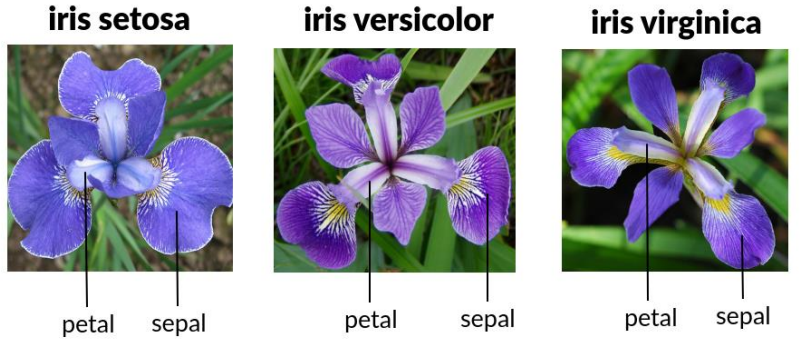
Online Fraud Detection

1. Introduction

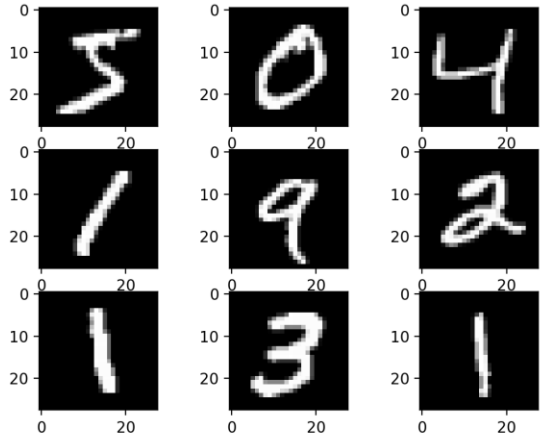
Examples:



Sentiment Analysis



Iris Classification



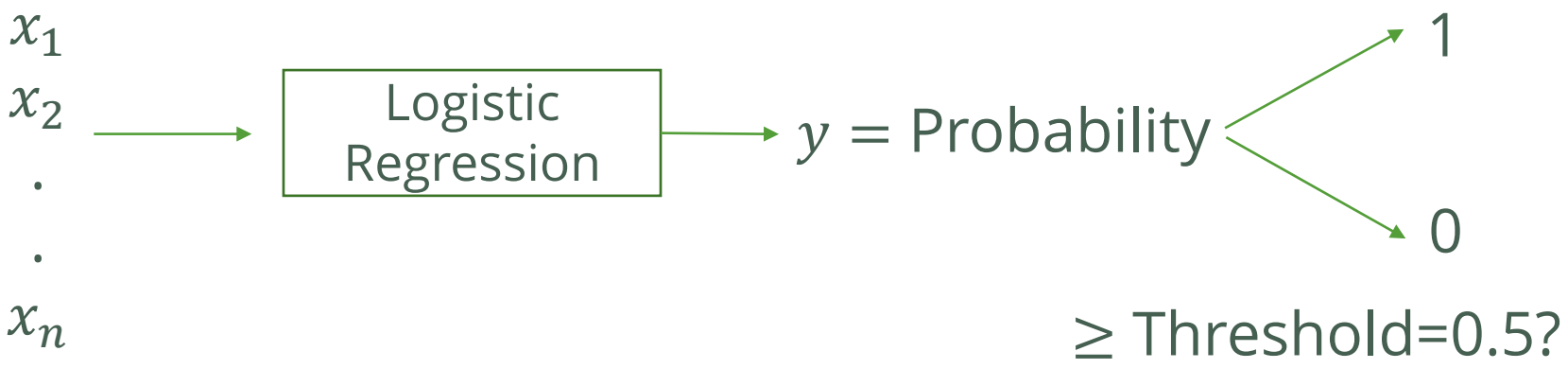
Digits Classification

1. Logistic Regression

- **Logistic Regression** is a popular statistical model which is used in **classification problem** to identify the class of a given observation by predicting the output value of categorical variable.
- It mostly deals with **binary/binomial classification** whose output value is binary value, which can be either 0 or 1 (true or false). However, it can be used for **multiclass/multi-binomial classification**.
- In this lecture, we will focus mainly on binary classification to simplify the problem.

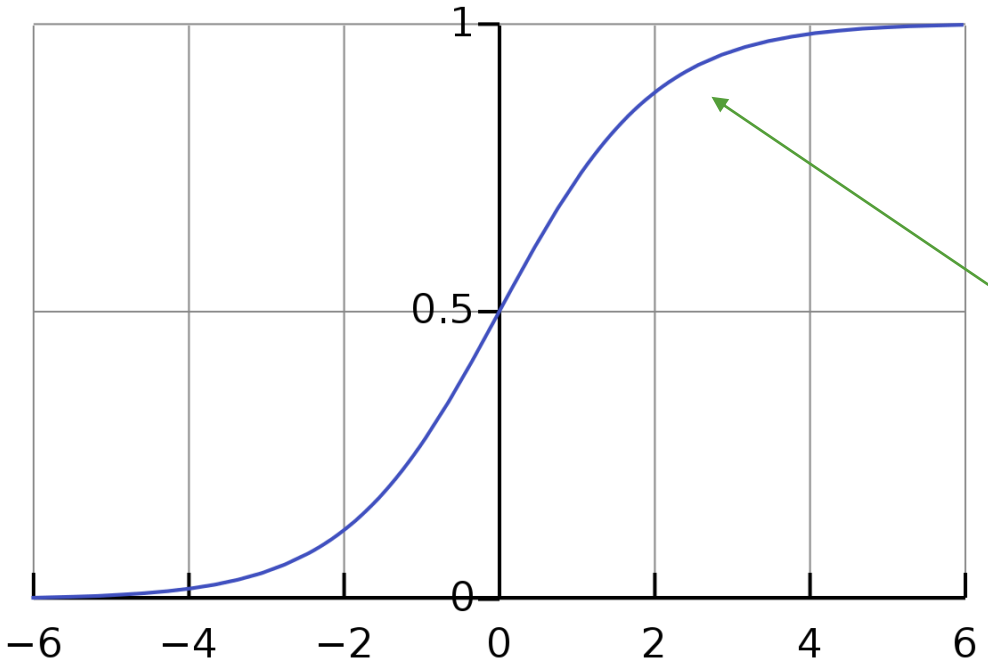
1. How does it work?

- Logistic Regression measures the relationship between the dependent variable (the label, what we want to predict) and one or more independent variables (the input features) by **estimating** the **probability** using the underlying **logistic function**.
- These probability is then transformed into binary value in order to actually make a prediction by using a **threshold**.



1. Sigmoid Function

Sigmoid Function (or **Logistic Function**) is an **S-sharped curve** function that can takes any real-valued number and maps it into a value between **range of 0 and 1**, but never exactly at those limit.

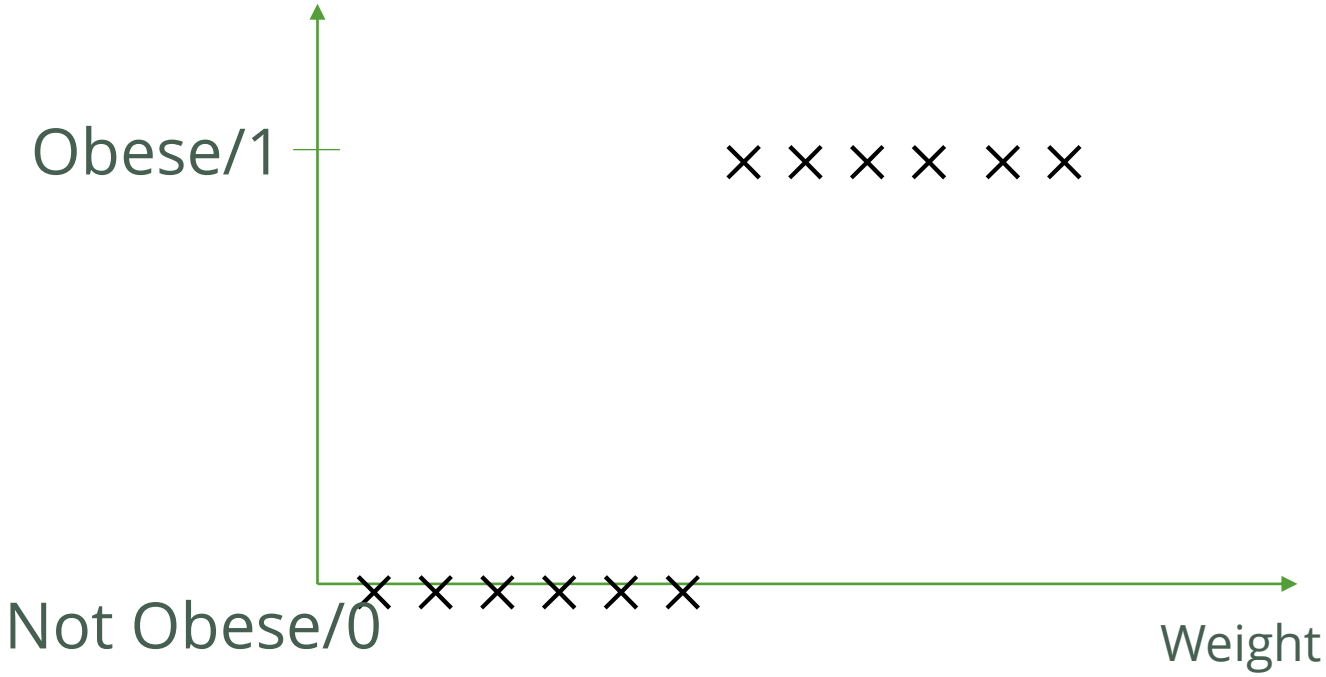


$$g(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid Function

1. Why not Linear Regression?

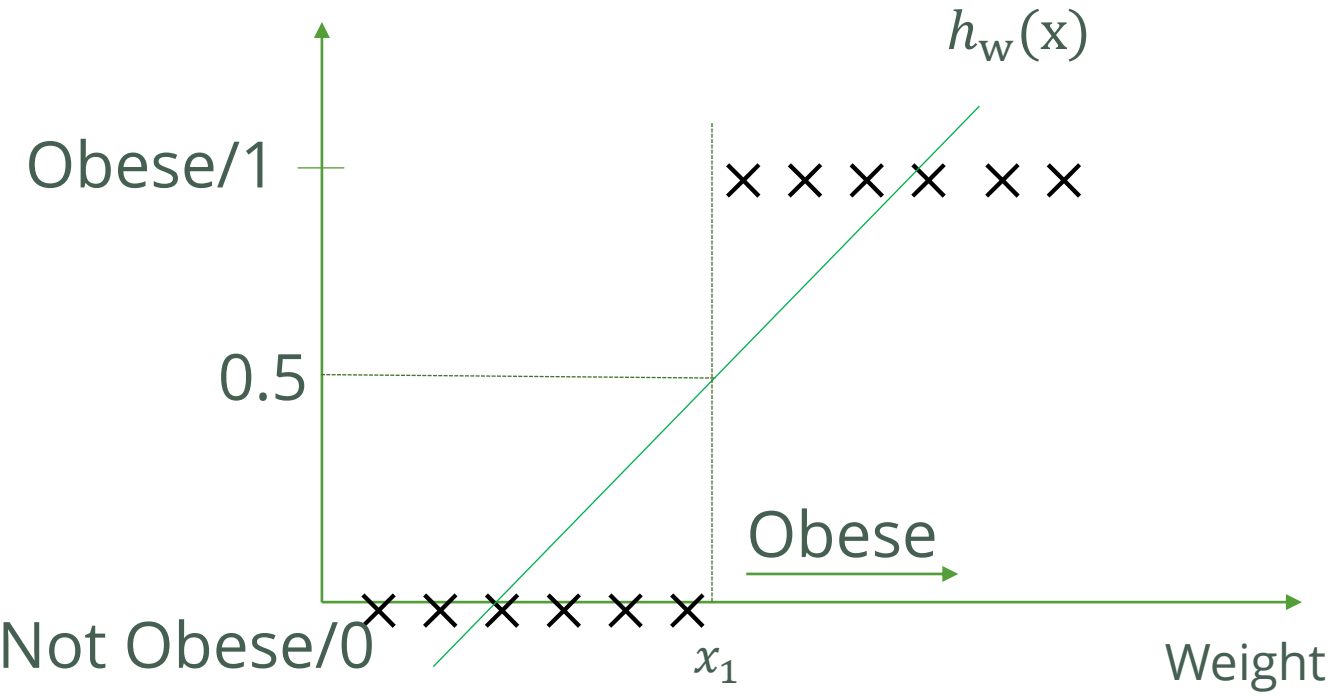
Example: Human Obesity Prediction



- How to use linear regression to solve classification when its prediction values are continuous and not bound between 0 and 1?

1. Why not Linear Regression?

Example: Human Obesity Prediction



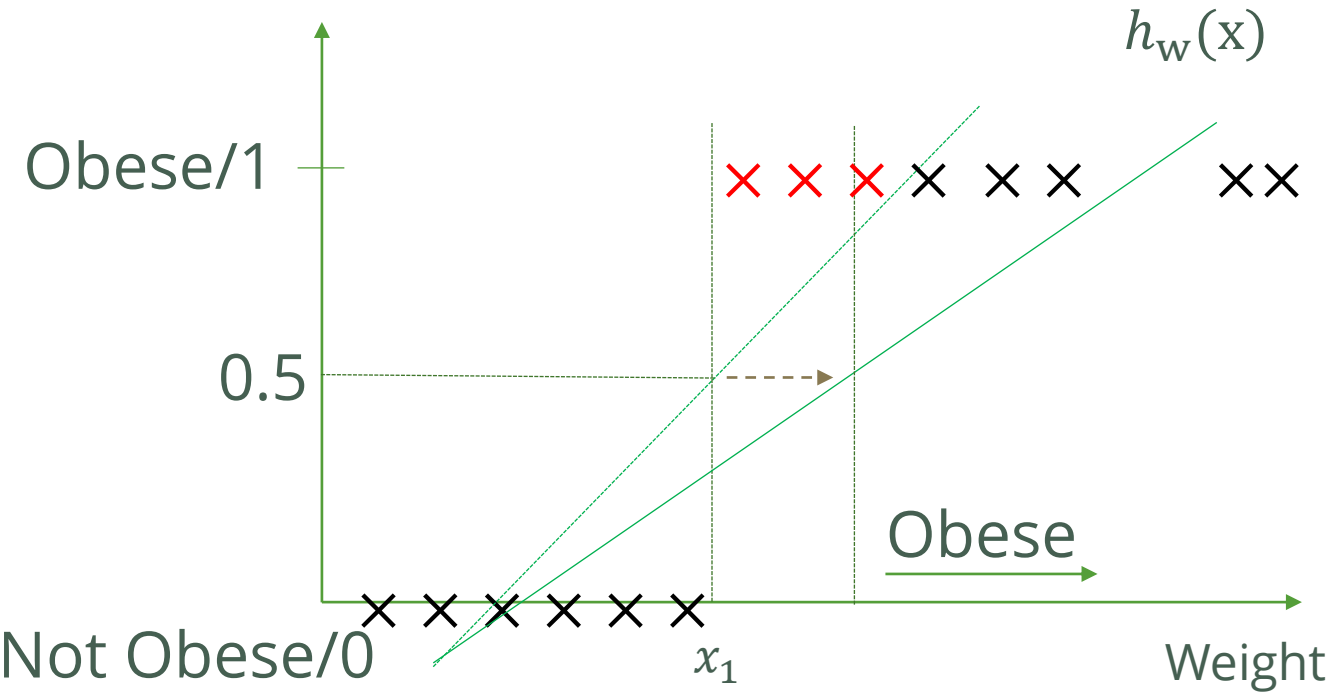
- To deal with classification, we can set a threshold of output value e.g. 0.5.

$$\text{if} \quad h_w(x) \geq 0.5 \implies \text{Obese/1}$$

$$\text{if} \quad h_w(x) < 0.5 \implies \text{Not Obese/0}$$

1. Why not Linear Regression?

Example: Human Obesity Prediction



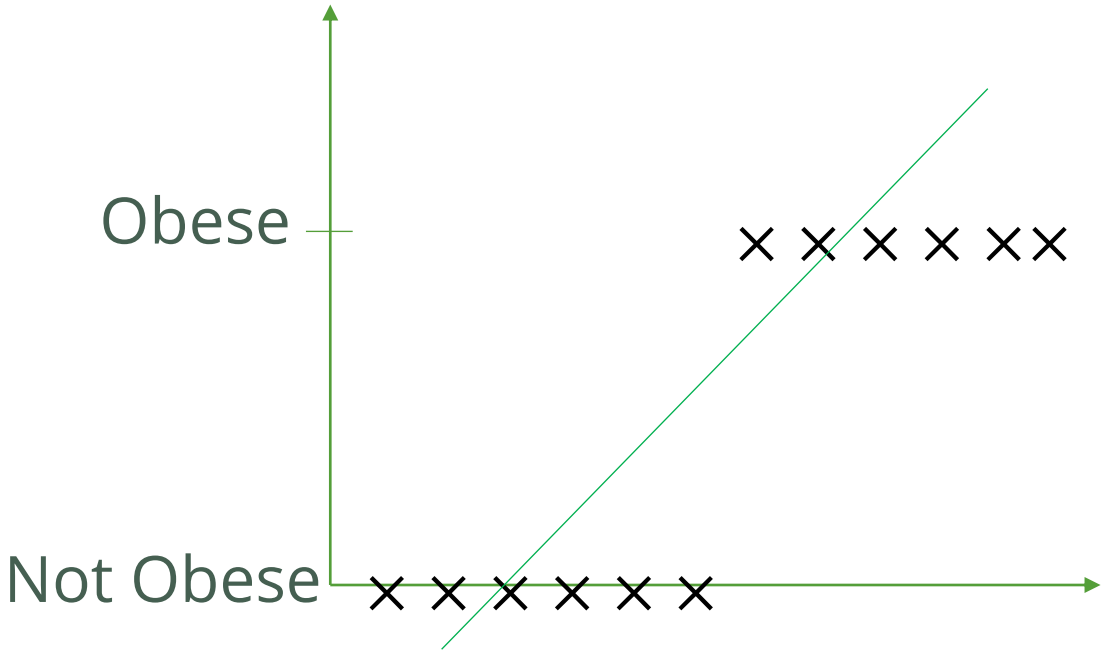
- Linear regression is not robust as the predictions have high error rate when there are the examples on the extreme right as this learning algorithm is sensitive to outliers.
- The red examples are classified as "Not Obese" while they are "Obese".

1. Linear Regression vs. Logistic Regression

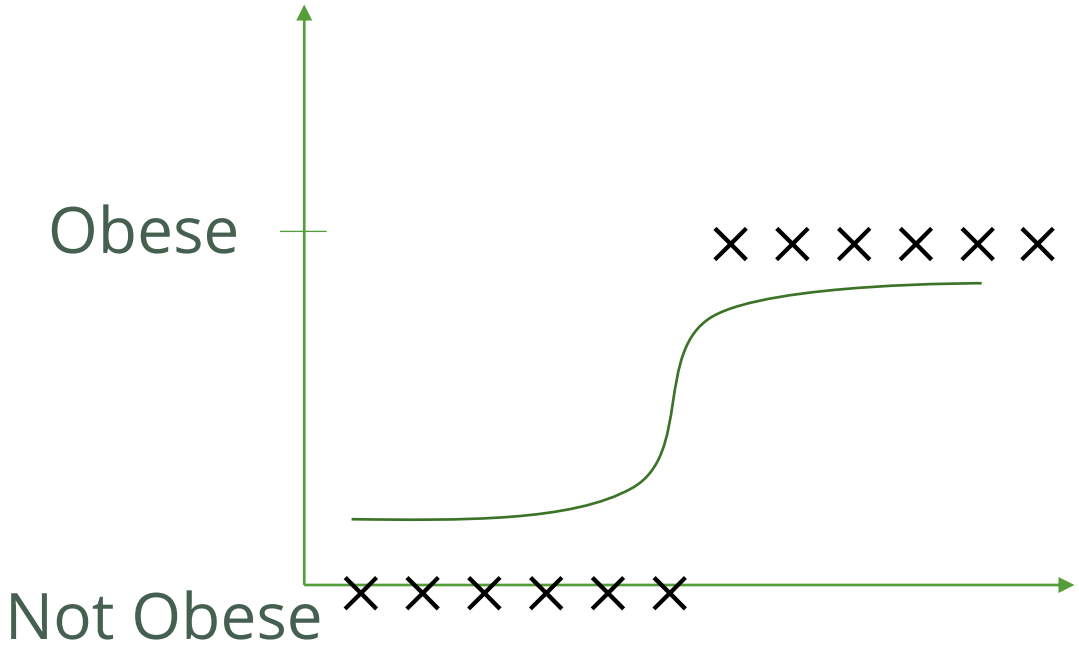
Linear Regression	Logistic Regression
Target variable is an **continuous**	Target variable is a **discrete**
Input variables have any measurement level	Input variables have any measurement level
Require linear relationship among dependent and independent variable	Not necessary for linear relationship between dependent and independent variable
Independent variables can be correlated with each other	Independent variables must not be correlated with each other
Predicted value are the mean of the target variable at the given values of the input variables	Predicted values are the probability of a particular levels of the target variable at the given values of input variables

1. Linear Regression vs. Logistic Regression

Example: Human Obesity Prediction - Logistic regression is more robust than linear regression.



Linear Regression



Logistic Regression

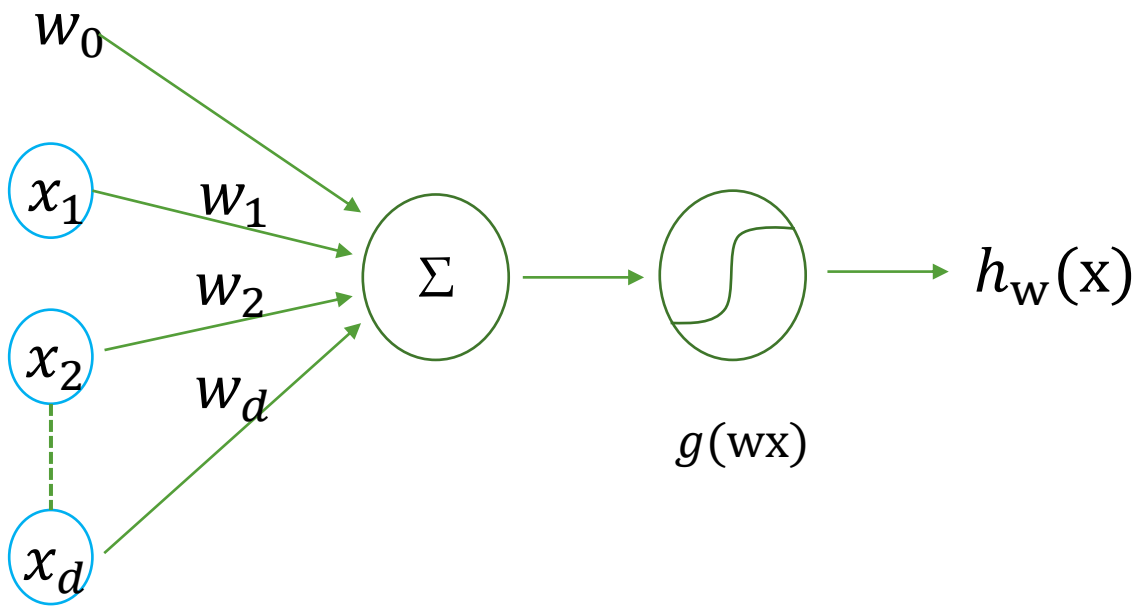
2. Hypothesis

Logistic Regression of d input variables $\mathrm{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ can be represented by the following hypothesis. It is derived by applying sigmoid function to the sum weighed of the input variables to output the probability.

$$h_{\mathrm{w}}(\mathrm{x}) = g(\mathrm{wx}), \qquad g(x) = \frac{1}{1 + e^{-x}}$$

$$\Longrightarrow h_{\mathrm{w}}\,(\mathrm{x}) = \frac{1}{1 + e^{-\mathrm{wx}}}$$

$$= P(y = 1 | \mathrm{x}, \mathrm{w})$$

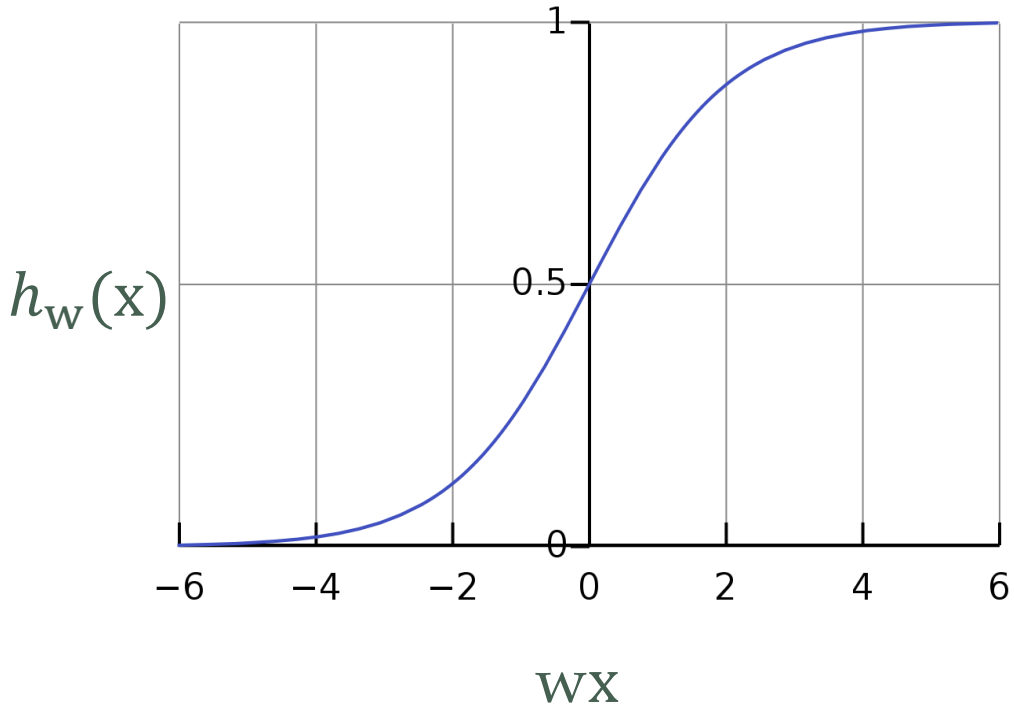


2. Hypothesis

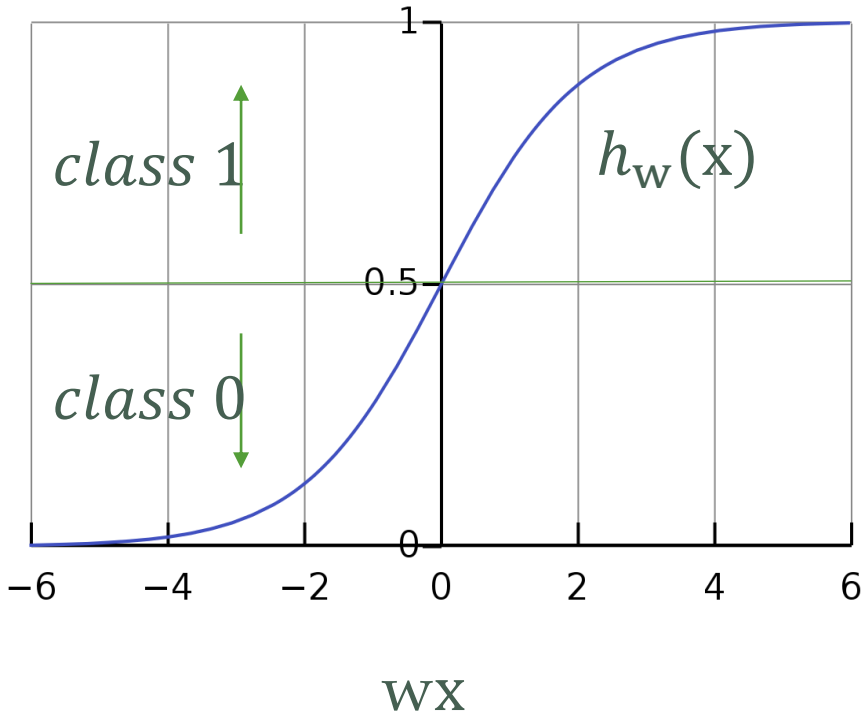
The following figure shows that $h_{\mathrm{w}}(\mathrm{x})$ is greater than or equals 0.5, when the wx is greater than or equals 0; and $h_{\mathrm{w}}(\mathrm{x})$ is less than 0.5 whenever, wx is less than 0.

$$h_{\mathrm{w}}(\mathrm{x}) = \frac{1}{1 + e^{-w\mathrm{x}}}$$

$$h_{\mathrm{w}}(\mathrm{x}) = \begin{cases} \geq 0.5, & if\ w\mathrm{x} \geq 0 \\ < 0.5, & if\ \ w\mathrm{x} < 0 \end{cases}$$



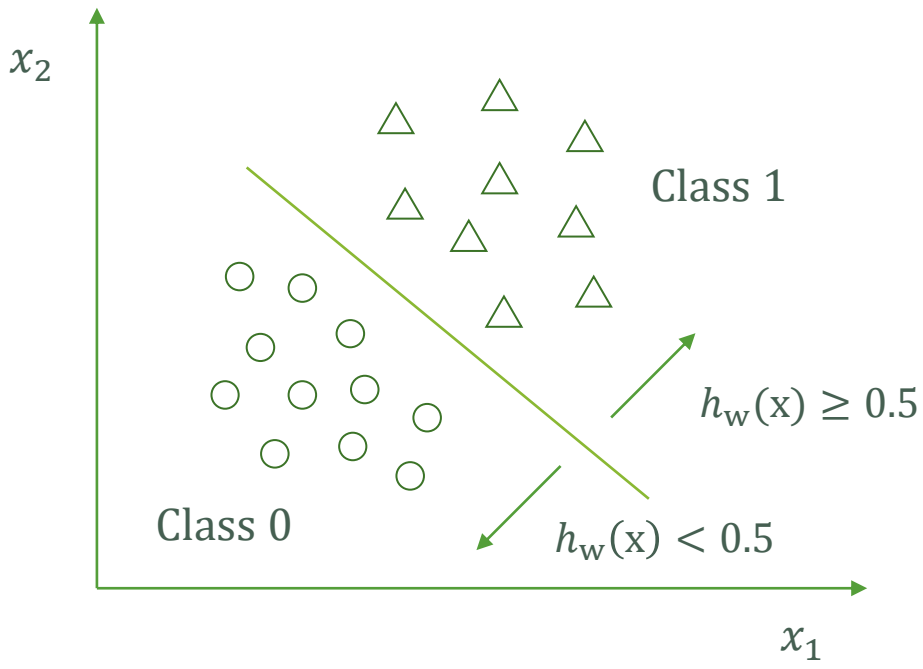
3. Decision Boundary



- In order to map $h_w(x)$ to discrete value of either 0 or 1, a threshold value of **0.5** is defined as a tipping point above which observations will be classified into class 0 – negative class or class 1 – positive class.

$$h_w(x) = \begin{cases} \geq 0.5 & \rightarrow class\ 1 \\ < 0.5 & \rightarrow class\ 0 \end{cases}$$

3. Decision Boundary



- This figure shows the decision boundary of Logistic Regression with two input variables $x = (x_1, x_2)$.
- All the observations which are classified into class 1, locates above the decision line, while the observations of class 0 are below the decision line.

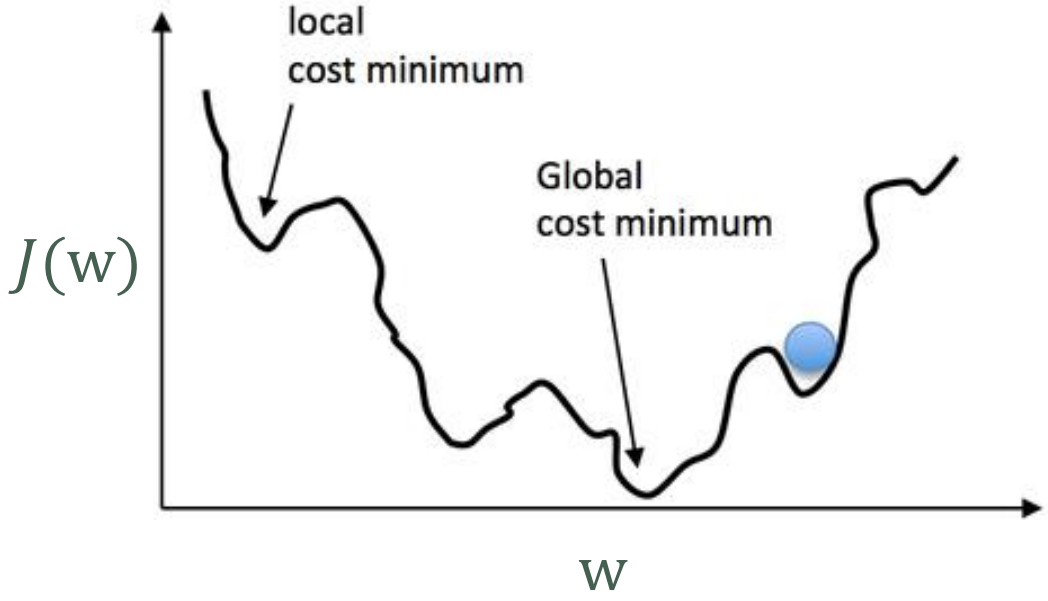
$$h_w(x) = \begin{cases} \geq 0.5 & \rightarrow class\ 1 \\ < 0.5 & \rightarrow class\ 0 \end{cases}$$

4. Cost Function

In logistic regression, we cannot use **Mean Squared Error** (MSE) as the non-linearity of the cost function $J(w)$, due to the sigmoid function. It results MSE a **non-convex function** with many local minimums. In such case, gradient descent cannot be used to learn the parameters.

$$J(\mathrm{w}) = \frac{1}{2n}\sum_{i=1}^{n} \left(h_{\mathrm{w}}(\mathrm{x}_i) - y_i\right)^2$$

$$h_{\mathrm{w}}(\mathrm{x}_i) = \frac{1}{1 + e^{-w\mathrm{x}_i}}$$



4. Cost Function

- Instead of MSE, we use a cost function, called **Cross-Entropy Loss**, also known as **Negative Log-likelihood**, which is a convex function and can be minimized by gradient descent.

- For a single training instance x_i, Cross-Entropy Loss is defined as follows:

$$J(\mathrm{w}) = -[y_i \log h_{\mathrm{w}}(\mathrm{x}_i) + (1 - y_i)\log(1 - h_{\mathrm{w}}(\mathrm{x}_i))]$$

- For the whole training set of n instances, it is the average of overall costs.

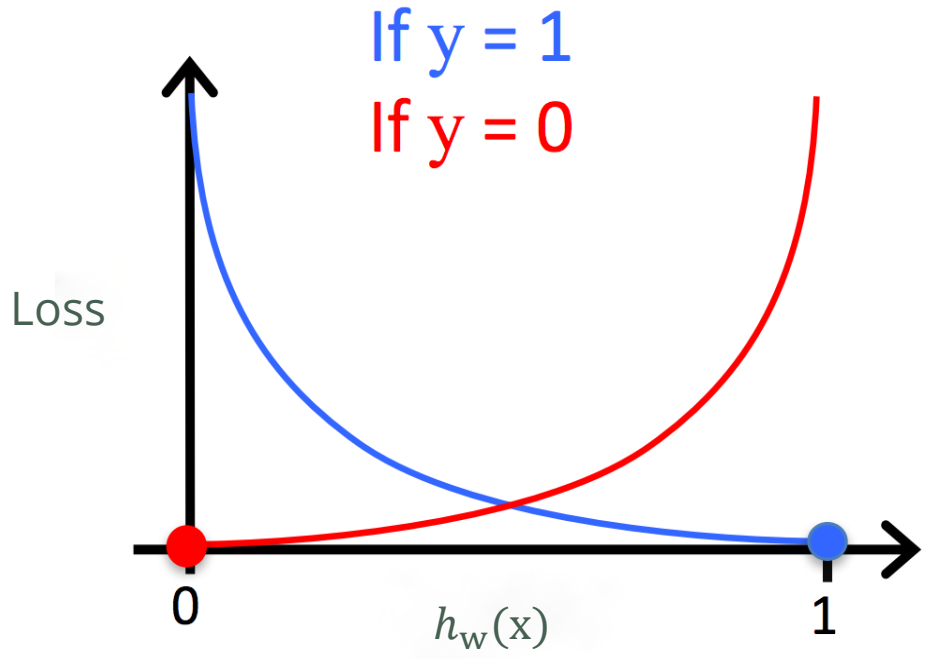
$$J(\mathrm{w}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log h_{\mathrm{w}}(\mathrm{x}_i) + (1 - y_i)\log(1 - h_{\mathrm{w}}(\mathrm{x}_i))]$$

4. Cost Function – Intuition

- Cross-Entropy Loss aims to find parameter $\hat{w}$ so that the model estimates high probabilities for the positive instances ($y = 1$) and low probabilities for the negative instances ($y = 0$).
- Cross-Entropy Loss is composed of two convex functions with respect to each class.

$$J(\mathrm{w}) = -[y_i \log h_{\mathrm{w}}(\mathrm{x}) + (1 - y_i)\log(1 - h_{\mathrm{w}}(\mathrm{x}))]$$

$$J(\mathrm{w}) = \begin{cases} -\log h_{\mathrm{w}}(\mathrm{x}) & if\ y = 1 \\ -\log\big(1 - h_{\mathrm{w}}(\mathrm{x})\big) & if\ y = 0 \end{cases}$$



5. Gradient Descent

- Given n examples $\{(x_i, y_1), (x_2, y_2), .., (x_n, y_n)\}$ such that $x_i \in \mathbb{R}^d$, the cost function:

$$J(w) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(h_w(x_i)) + (1 - y_i)\log(1 - h_w(x_i)]$$

- The gradient or partial derivative of cost function with respect to each parameter w_i:

$$\frac{\partial}{\partial w_i}J(w) = \frac{1}{n}\sum_{i=1}^{n}(h_w(x_i) - y_i)x_i$$

5. Gradient Descent

- The batch gradient descent algorithm of logistic regression is outlined as follows:

Initialize $w_0, \ldots, w_d$ with random values

For all data points:

make predictions – compute $h_{\mathrm{w}}(\mathrm{x_i})$

For each weight w_i:

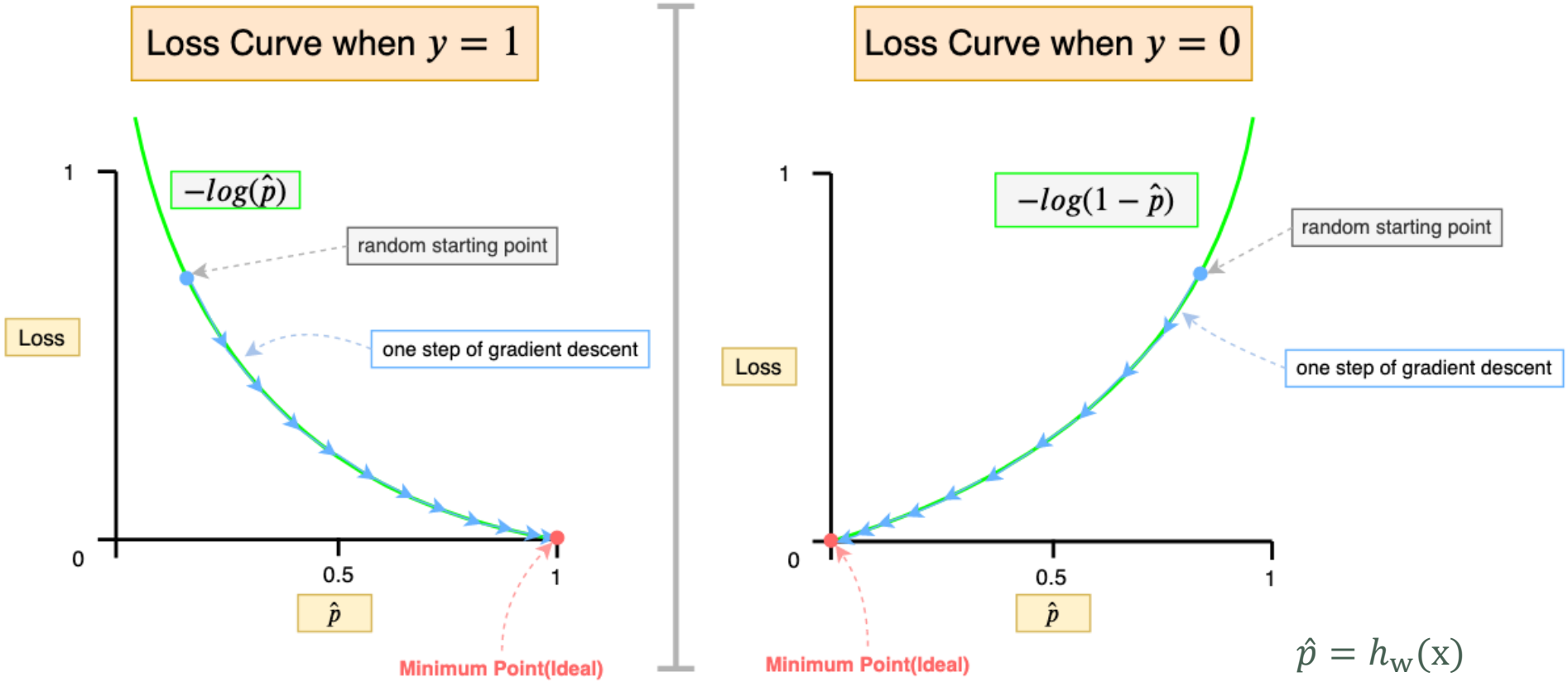
compute the gradient and update new weight

$$w_{\mathrm{i}} = w_{\mathrm{i}} - \frac{\eta}{n}\sum_{i=1}^{n}(h_{\mathrm{w}}(\mathrm{x_i}) - y_i)\, x_{\mathrm{i}}$$

Until weights converges

5. Gradient Descent

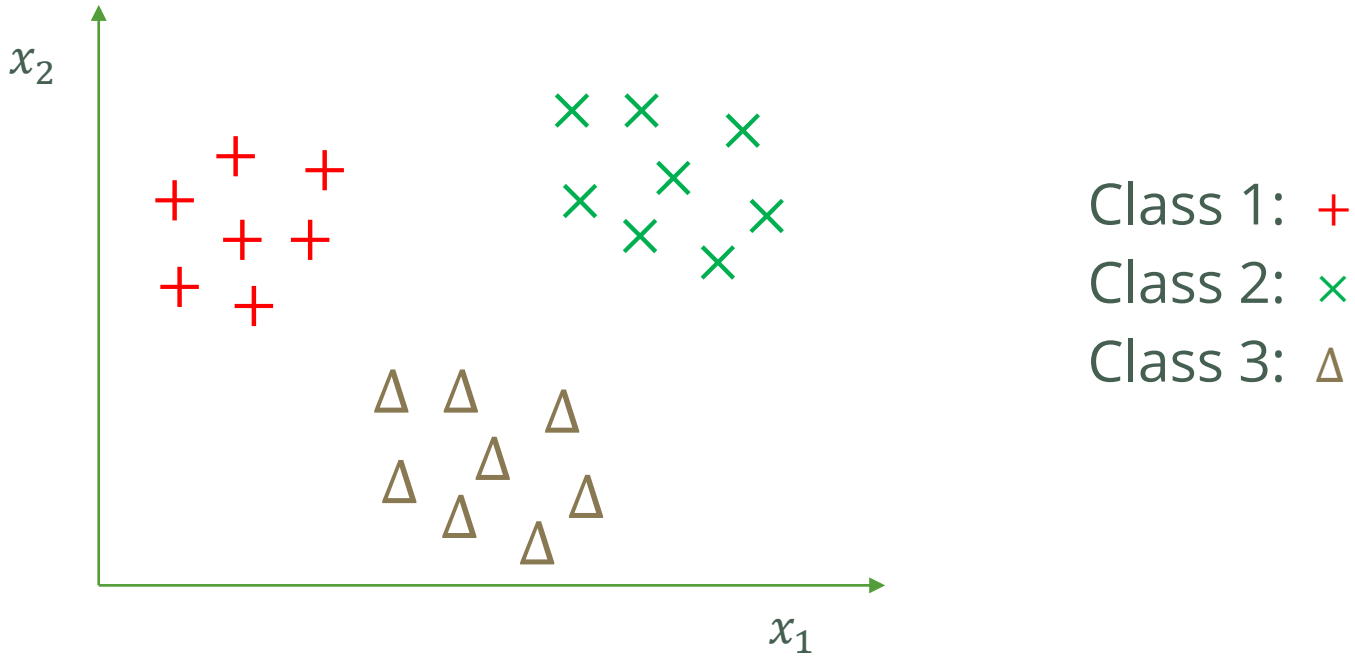
- The batch gradient descent algorithm of logistic regression is outlined as follows:



$$\hat{p} = h_{\mathrm{w}}(\mathrm{x})$$

6. Multi-Class Classification

- What is the intuition behind the multi-class classification in Logistic Regression?

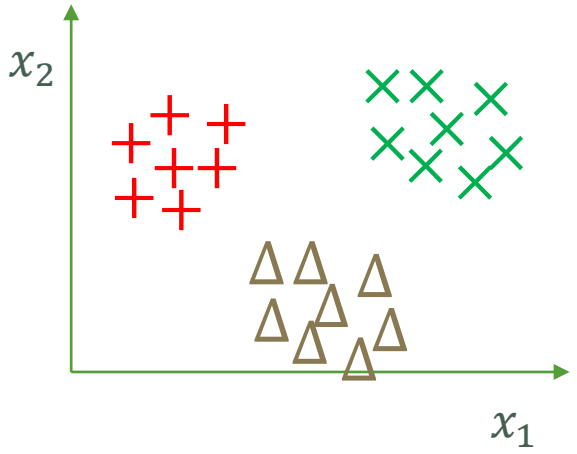


6. One vs. Rest Classification

- Logistic Regression is mainly focus on binary classification. However, it can used to solve with multi-class classification by using a technique, called ” **One vs. Rest Classification**”.
- ”One vs. Rest Classification” strategy involves training a single classifier per class, with the samples of that class as positive samples and all the other samples as negatives. The predictions are made using the model that is the most confident.

6. One vs. Rest Classification

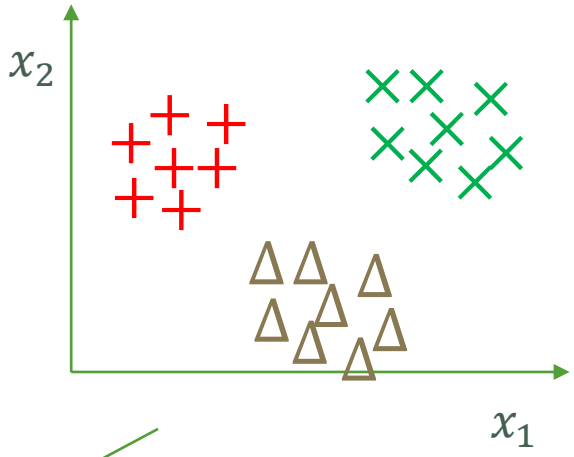
Class 1: +
Class 2: ×
Class 3: Δ



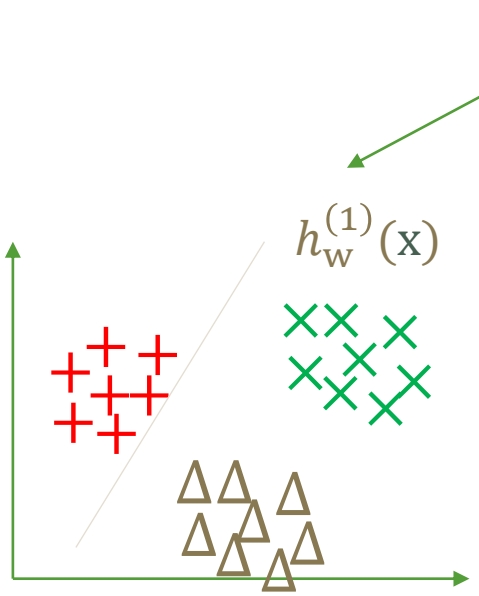
$$h_{\mathrm{w}}^{(k)}(\mathrm{x}) = P(y = k|\mathrm{x}, \mathrm{w})$$

6. One vs. Rest Classification

Class 1: +
Class 2: ×
Class 3: Δ

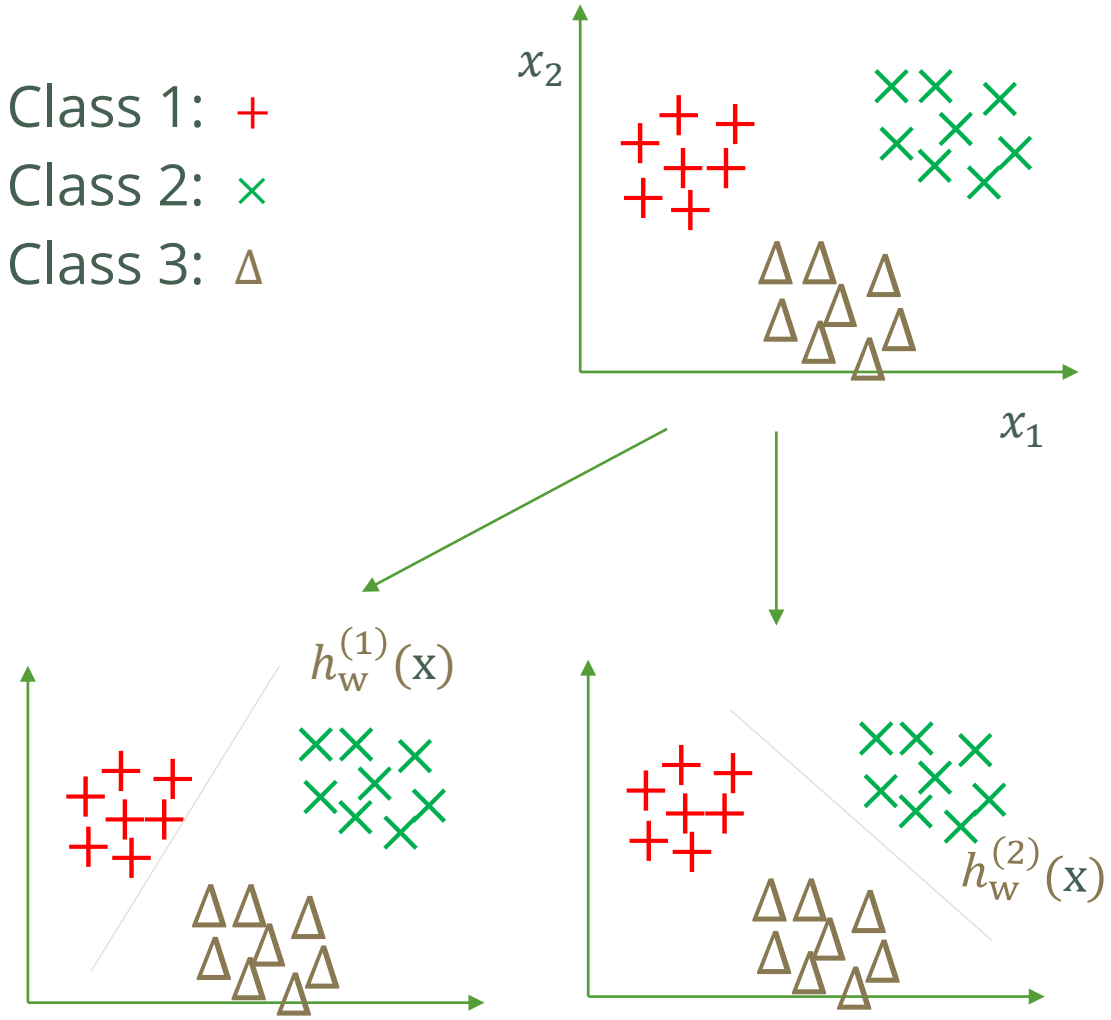


$$h_{\mathrm{w}}^{(k)}(\mathrm{x}) = P(y = k|\mathrm{x}, \mathrm{w})$$



6. One vs. Rest Classification

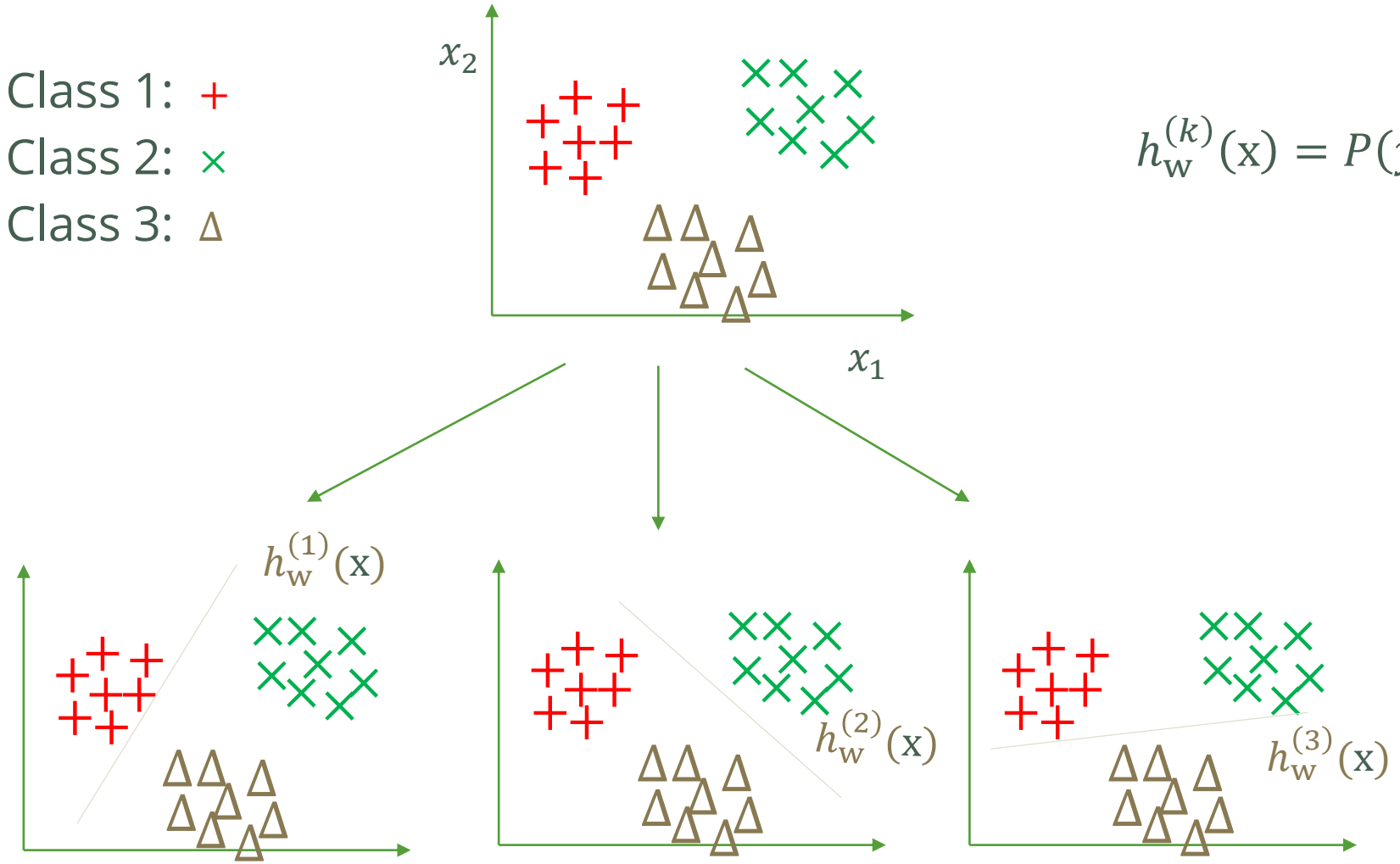
Class 1: +
Class 2: ×
Class 3: Δ



$$h_w^{(k)}(x) = P(y = k|x, w)$$

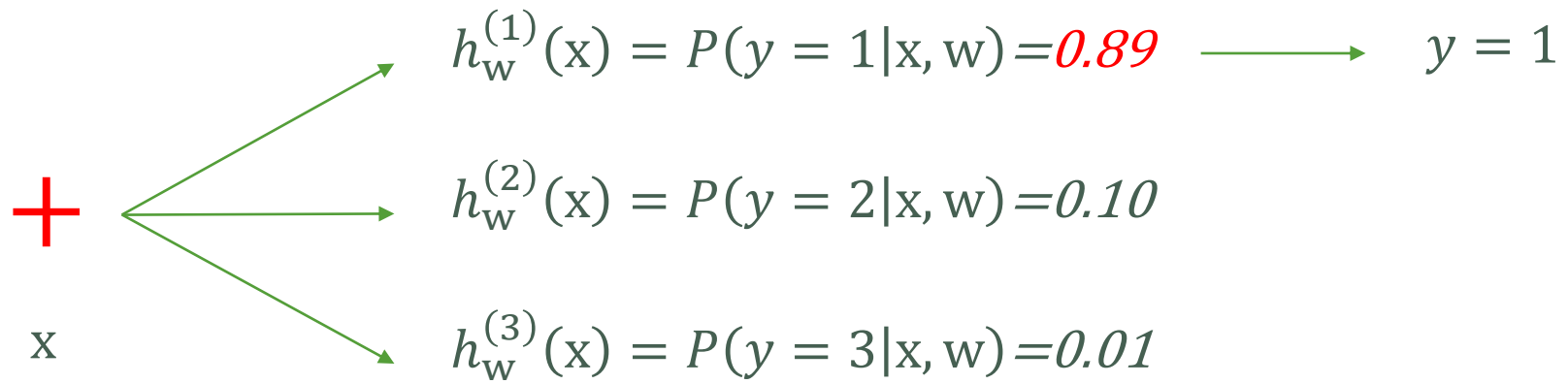
6. One vs. Rest Classification

Class 1: +
Class 2: ×
Class 3: Δ



$$h_w^{(k)}(x) = P(y = k|x, w)$$

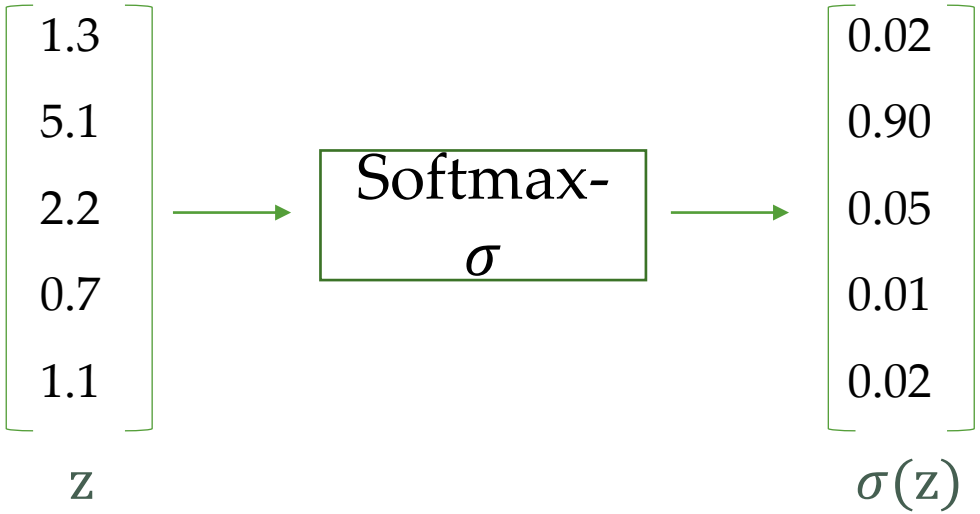
6. One vs. Rest Classification



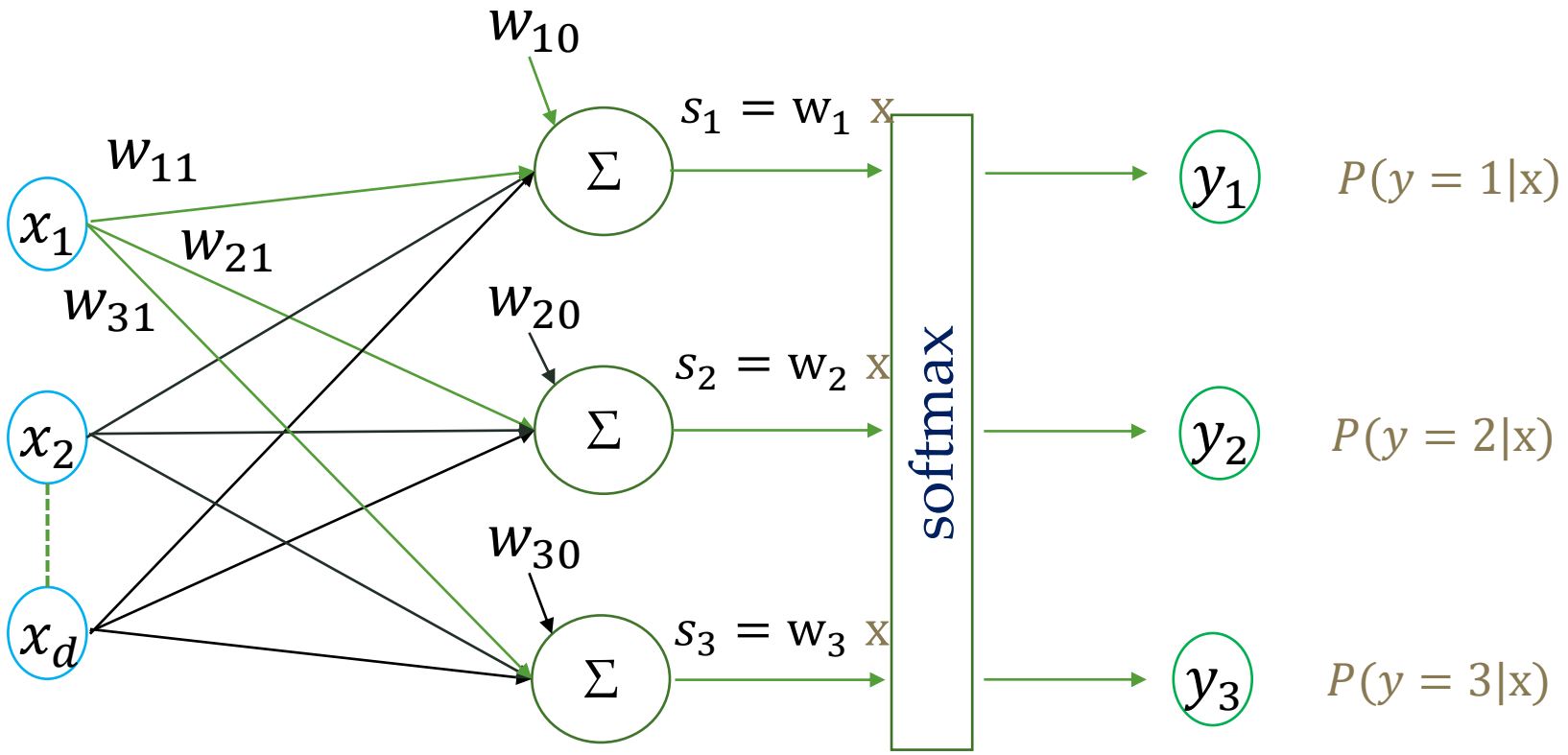
6. Softmax Regression

- Logistic Regression can be generalized to support multiple classes directly, without having to train and combine multiple binary classifiers. This is called **Softmax Regression** or **Multinomial Logistic Regression**.
- The idea is quite simple: when given an instance x, Softmax Regression first computes a score $s_k(\mathrm{x})$ for each class k, then estimates the probability of each class by applying the **softmax function** (also called the normalized exponential) to the scores.

$$\sigma(\mathrm{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$$



6. Softmax Regression



$$P(y = k) = \sigma\big(s(\mathrm{x})\big)_k = \frac{e^{s_k(\mathrm{x})}}{\sum_{j=1}^{K} e^{s_j(\mathrm{x})}} \quad , \quad J(w) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} \log\left(P^{(i)}(y = k)\right)$$

7. Pros & Cons

- Pros:
  - Easy to implement and interpret results
  - Inexpensive computation
  - Not require feature scaling
- Cons:
  - Not able to handle a large number of categorical output variable
  - Vulnerable to overfitting
  - Cannot solve non-linear data