Logistic Regression

Data Science Dojo

Classification

- Two-class (binary) classification problem

  y : {0,1}   0: negative class

                   1: positive class

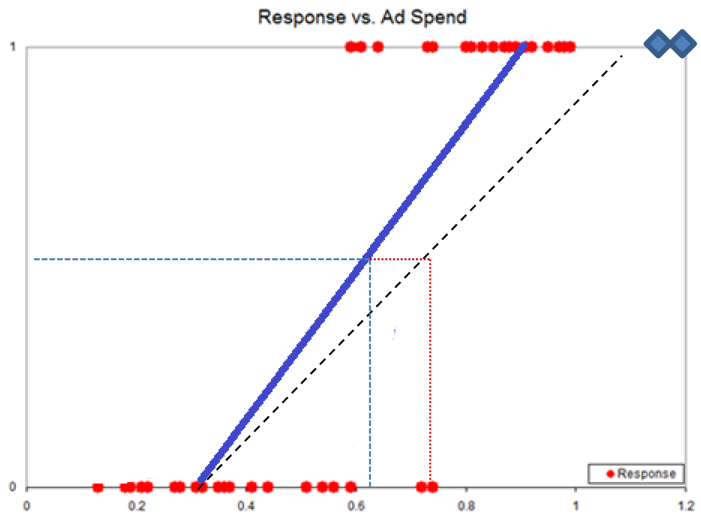
  - Example: Yes/No; Benign/Malignant ; Click/No click

- Multi-class classification problem

  y:{0,1,2,.......n}

  - Example: Grades (A, B, C); color (red, blue, green)

Two Class Classification



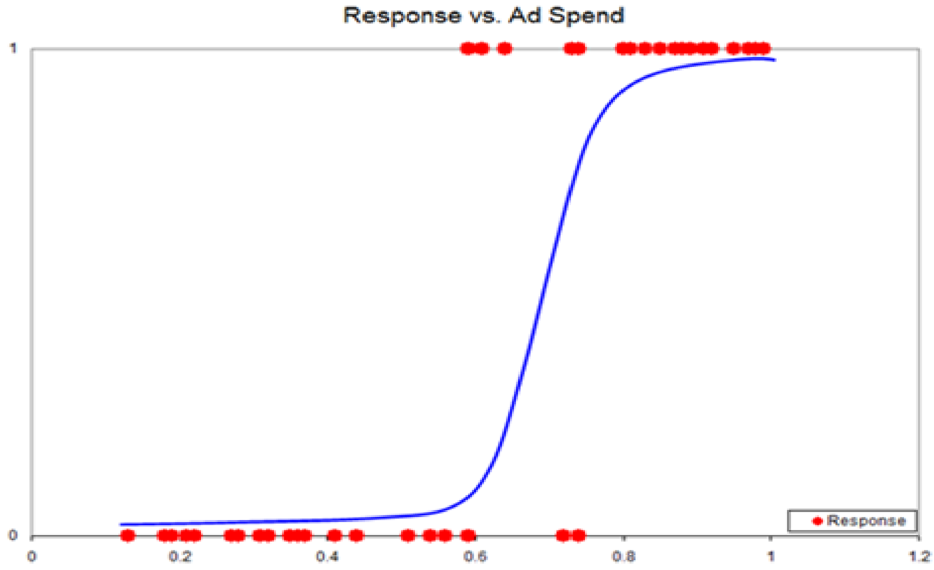
Classification: y= 0 or y=1

Decision rules: $h_\theta(x) \geqslant 0.5$; y = 1
$h_\theta(x) < 0.5$; y = 0

If we use linear regression on classification problem, we may observe

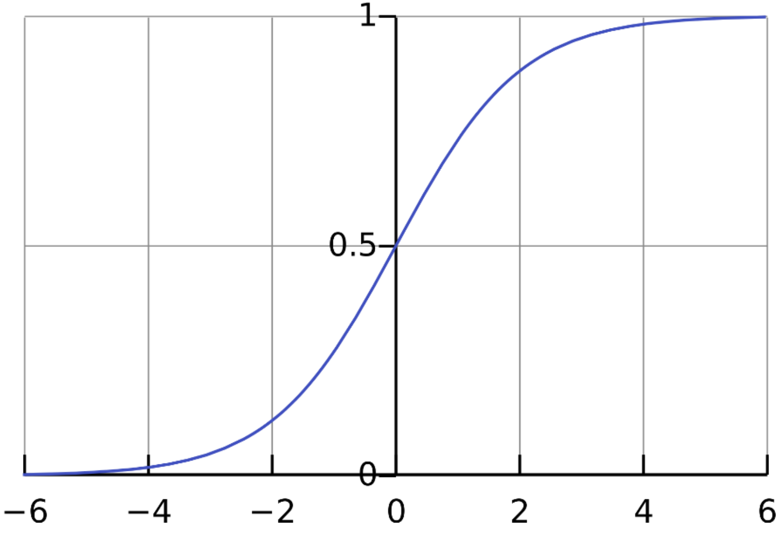
- Shift the decision rule line
- $h_\theta(x)$ can be > 1 or < 0

Logistic Regression



More reasonable function use for two-class classification problem.

Sigmoid Function



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

Hypothesis interpretation

Estimated probability that y=1 on x input

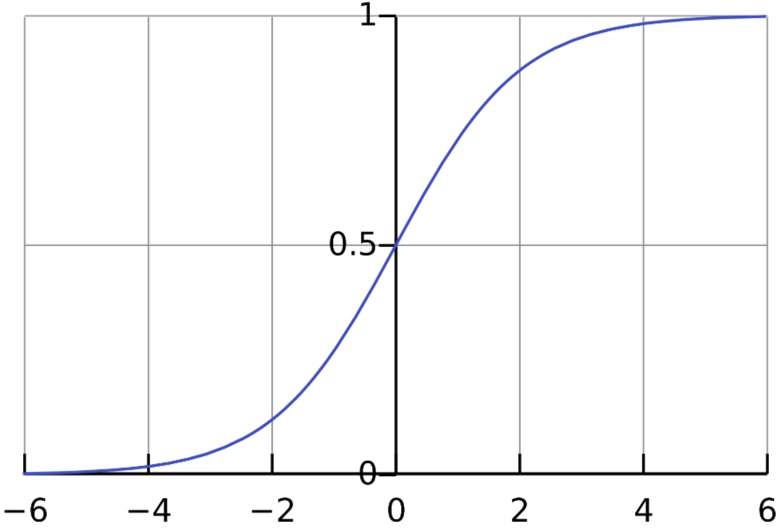
$P(y=1 | x; \theta_j)$

Because probabilities should sum to 1

$P(y=0 | x; \theta_j) = 1 - P(y=1 | x; \theta_j)$

datasciencedojo
unleash the data scientist in you

Sigmoid Function



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

y = 1 if $h_\theta(x) \geqslant 0.5$

y = 0 if $h_\theta(x) < 0.5$

0 asymptote for x $\longrightarrow$ - ∞

1 asymptote for x $\longrightarrow$ ∞

Cost Function for Logistic Regression

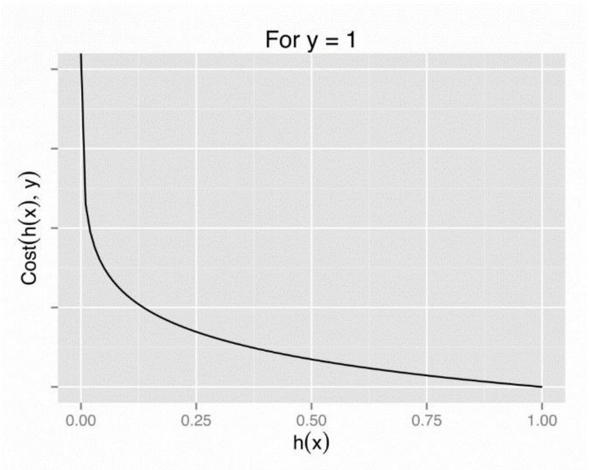
Average cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

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

Cost Function for Logistic Regression

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



If we predict $h_\theta(x) = 1$ and y = 1

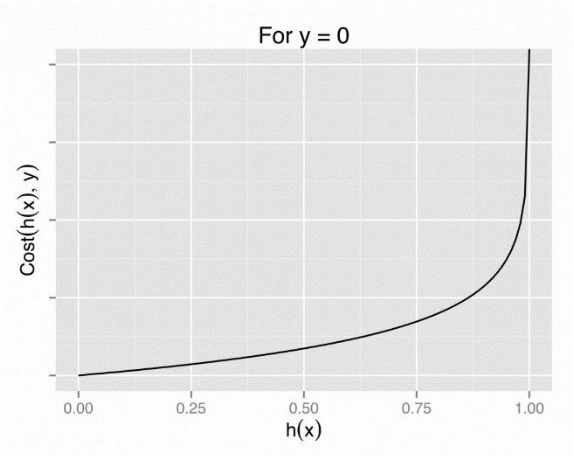
Cost function ⟶ zero

If we predict $h_\theta(x) = 0$ and y =1

Cost function ⟶ - ∞

Cost Function for Logistic Regression

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



If we predict h_θ(x) = 0 and y = 0

Cost function ⟶ zero

If we predict h_θ(x) = 1 and y =0

Cost function ⟶ ∞

Cost Function for Logistic Regression

Average cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

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

Simplified cost function, given we only have either y=0 or y=1

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

datasciencedojo

unleash the data scientist in you

Cost Function: Recap

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

Cost Function: $J(\theta) = -\dfrac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$

Parameters: θ_j

Goal: $\underset{\theta}{argmin}\ J(\theta)$

datasciencedojo
unleash the data scientist in you

Gradient Descent Algorithm

- We want to learn the values of θ that minimize $J(\theta)$
- Use a search algorithm that starts with an initial guess for θ and then changes θ to make $J(\theta)$ smaller
- Gradient descent starts with some initial θ and then performs an update for each value θ_j

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta).$$

Minimizing The Cost Function J(θ)

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

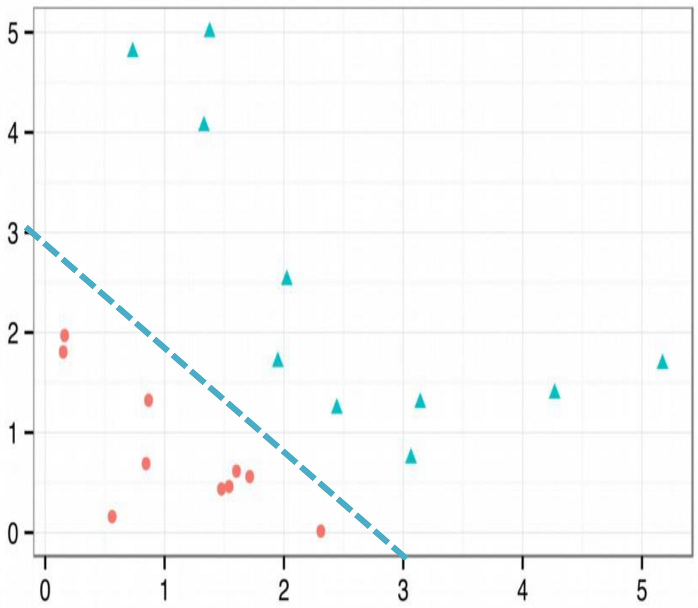
After derivative

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

$$\theta := \begin{bmatrix} tmp_0 \\ \vdots \\ tmp_n \end{bmatrix}$$

Repeat until converged

Decision Boundary: Linear



If $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

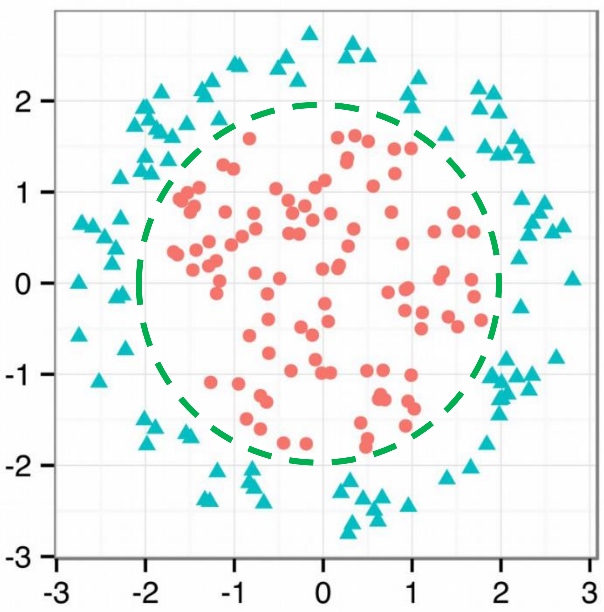
And $\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

Prediction y = 1 whenever

$$\begin{array}{rcl} & \theta^T x & \geq & 0 \\ \Leftrightarrow & -3 + x_1 + x_2 & \geq & 0 \\ \Leftrightarrow & x_1 + x_2 & \geq & 3 \end{array}$$

datasciencedojo
unleash the data scientist in you

Decision Boundary: Non-Linear



If $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$

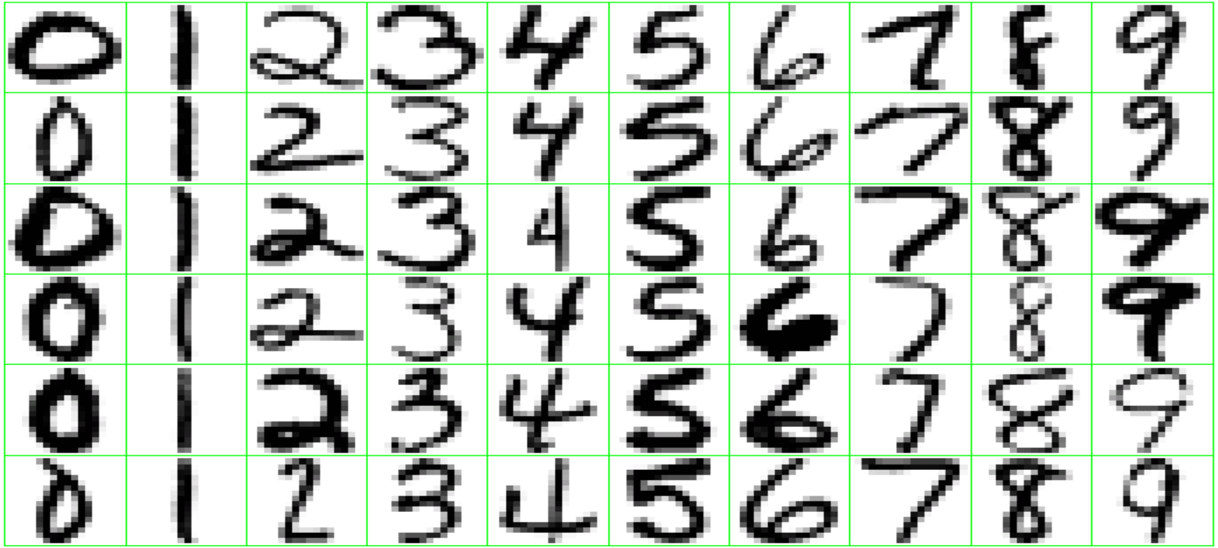
And $\theta = \begin{bmatrix} -2 & 0 & 0 & 1 & 1 \end{bmatrix}^T$

Prediction y = 1 whenever

$$x_1^2 + x_2^2 \geq 2$$

datasciencedojo
unleash the data scientist in you

Example: Handwritten Digit Recognition



datasciencedojo
unleash the data scientist in you

QUESTIONS