CAP 4630 – Logistic Regression

Instructor: Aakash Kumar

University of Central Florida

Logistic Regression

- **Introduction**:
  - Logistic regression is a supervised learning algorithm used for binary classification tasks.
  - It predicts the probability of a binary outcome (0 or 1) based on input features.
- **Examples of Classification**:
  - Email: Spam / Not Spam?
  - Online Transactions: Fraudulent (Yes / No)?
  - Tumor: Malignant / Benign?
- **Binary Outcomes**:
  - Class 0: Negative Class (e.g., benign tumor)
  - Class 1: Positive Class (e.g., malignant tumor)
- **Threshold for Prediction**:
  - Logistic regression uses a threshold (commonly 0.5):
    - If the probability >0.5, predict "1“
    - Otherwise, predict "0"

Comparison Between Linear and Logistic Regression

- **Linear Regression**:
  - Used when the **dependent variable is continuous** (e.g., predicting house prices or temperatures).
  - Fits a straight line to the data to predict a continuous outcome.

- **Logistic Regression**:
  - Used when the **dependent variable is binary** (e.g., spam detection, tumor classification).
  - Predicts probabilities of outcomes (0 or 1) using the logistic function.

Linear Regression

- Hypothesis Function

$$y = \theta_0 + \theta_1 \cdot x$$

- Cost Function

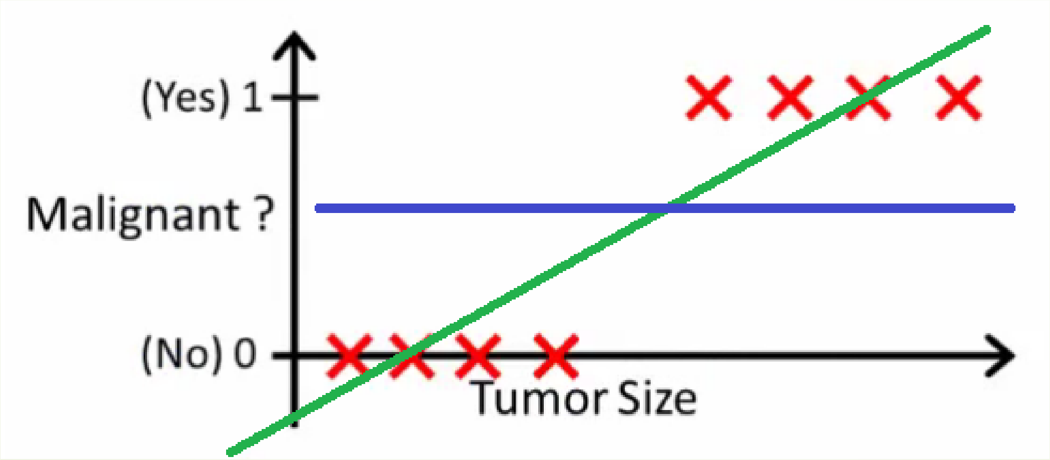
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

- Optimization/Gradient Descent (values of parameters)

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

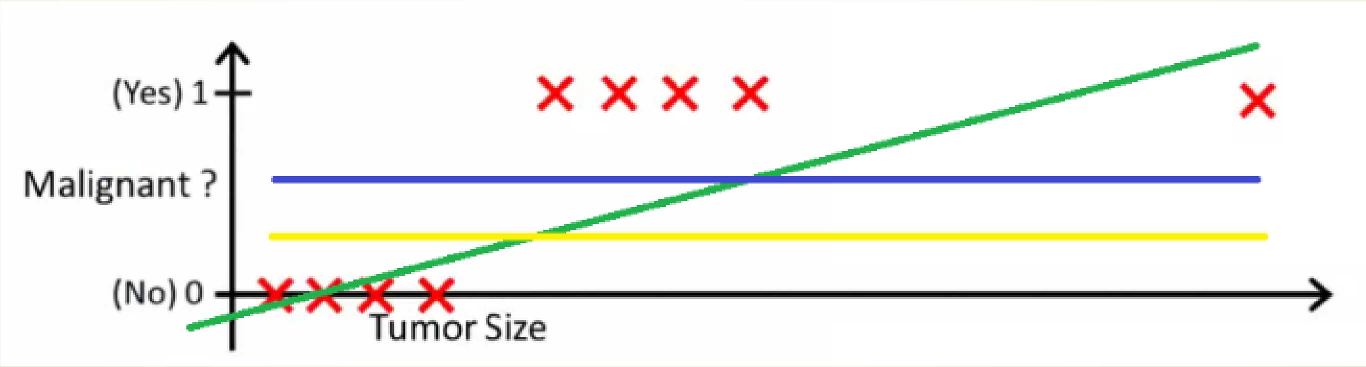
Comparison Between Linear and Logistic Regression

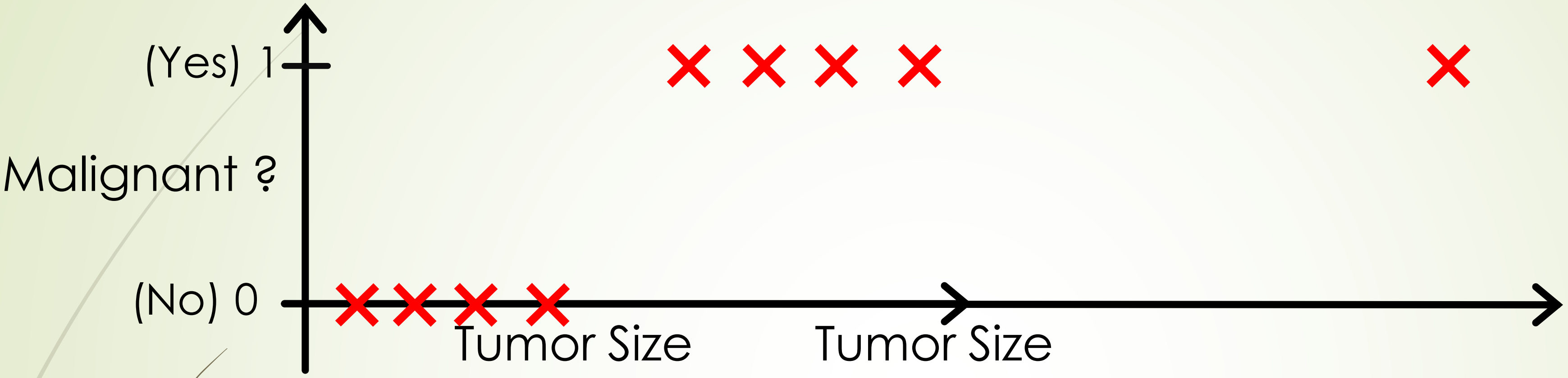
- The threshold is set at 0.5:
  - If h(x)>0.5, predict the tumor is **malignant (1)**.
  - If h(x)<0.5, predict the tumor is **benign (0)**.



Comparison Between Linear and Logistic Regression

- The **blue line** represents the original threshold (0.5), while the **yellow line** shows the new threshold, which is 0.2.
- To maintain accurate predictions, we had to lower the threshold.
- This demonstrates that **linear regression** is highly sensitive to outliers.
- Now, the model predicts correctly only when $h(x)>0.2$.





Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Classification: y = 0 or 1

$h_\theta(x)$ can be > 1 or < 0

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Sigmoid Function

- The sigmoid function is a key element in logistic regression and neural networks. It transforms input values (z) into outputs that lie between 0 and 1, making it useful for binary classification tasks.
- The formula for the sigmoid function is:

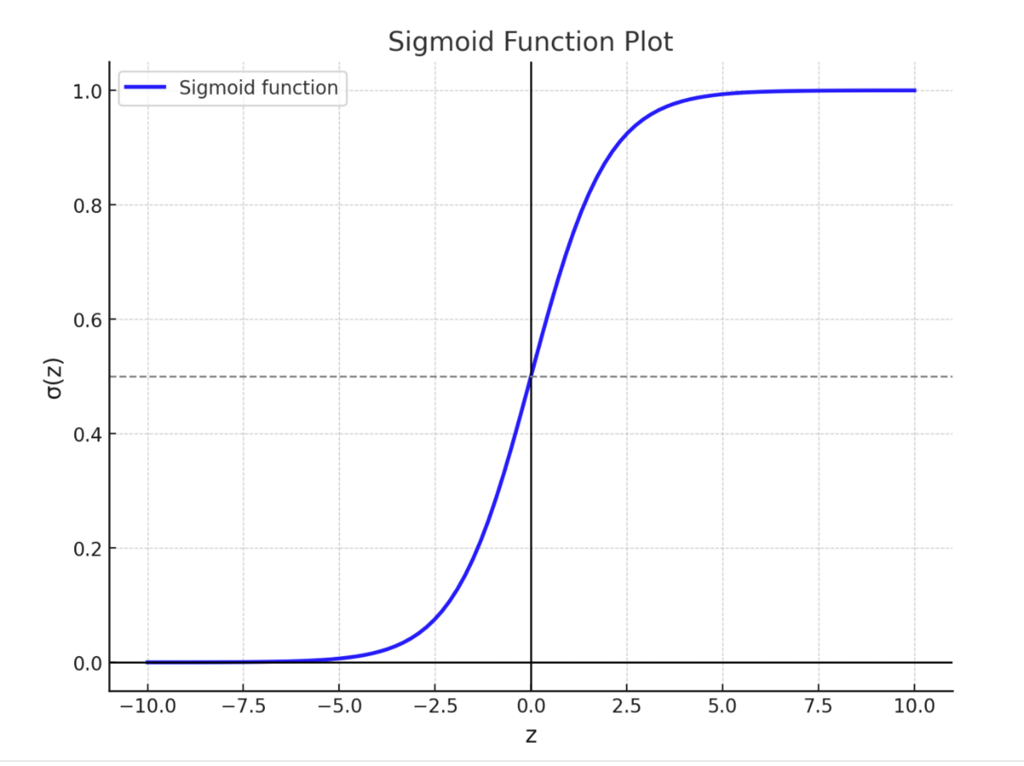
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Where:

- z is the input to the function, often representing a linear combination of inputs (e.g., $z = \theta^T X$ in logistic regression).
- e is the base of the natural logarithm.

Behavior:

- When $z \to +\infty$, $\sigma(z) \to 1$.
- When $z \to -\infty$, $\sigma(z) \to 0$.
- When $z = 0$, $\sigma(z) = 0.5$.



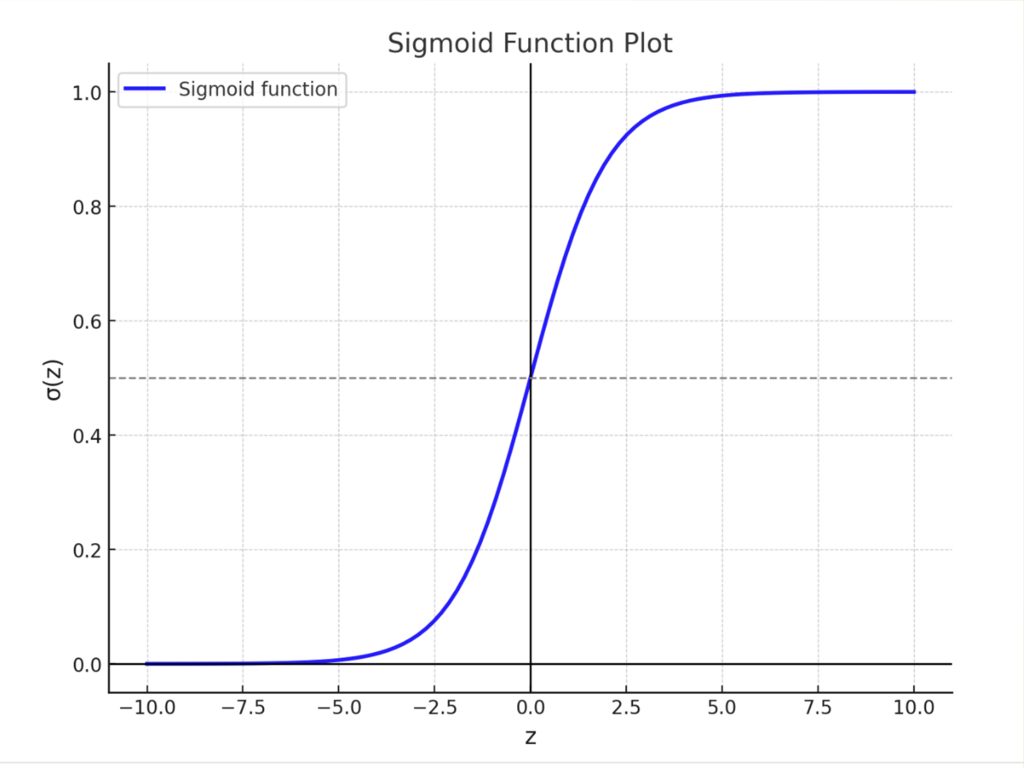
Hypothesis Function for Logistic Regression

Want $0 \leq h_\theta(x) \leq 1$

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

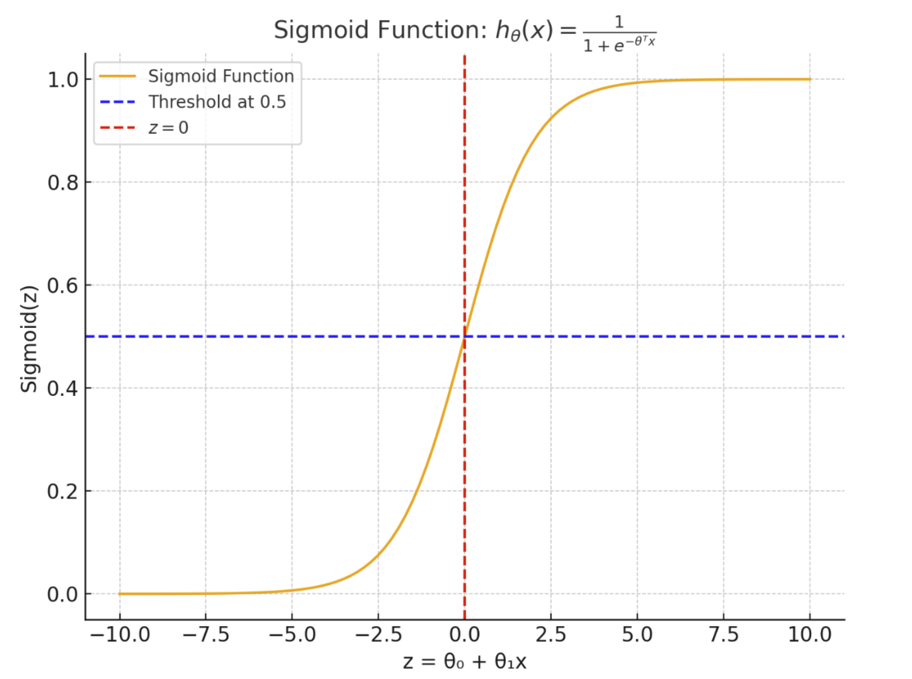
$$h_\theta(x) = g(\theta^T x)$$

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



Hypothesis Function

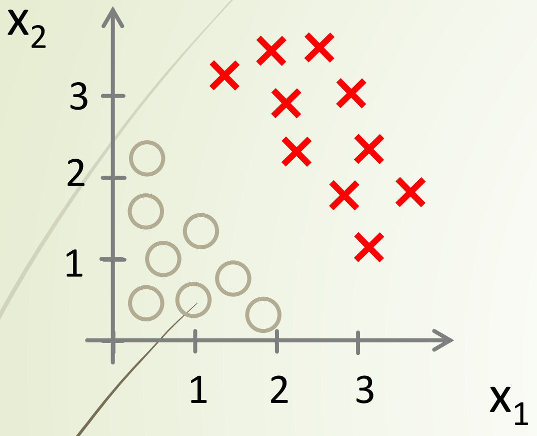
$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$



Explanation:

- θ_0 is the intercept (bias term).
- θ_1 is the weight for the feature x.
- The term $\theta_0 + \theta_1 x$ is the linear combination of the parameters and the input feature.
- The sigmoid function $\frac{1}{1+e^{-z}}$, where $z = \theta_0 + \theta_1 x$, transforms the output of the linear function into a value between 0 and 1, which can be interpreted as a probability.

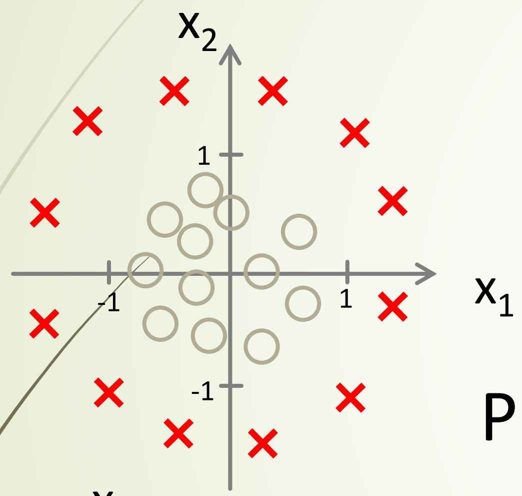
Decision Boundries



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

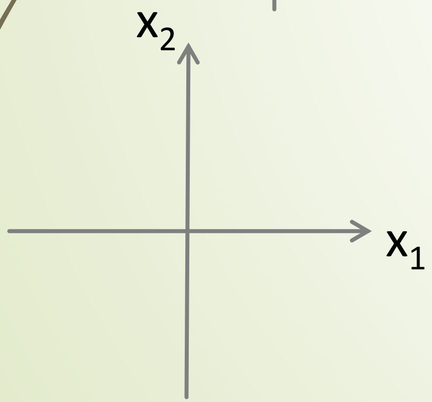
Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

Non-linear Decision Boundries



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

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$

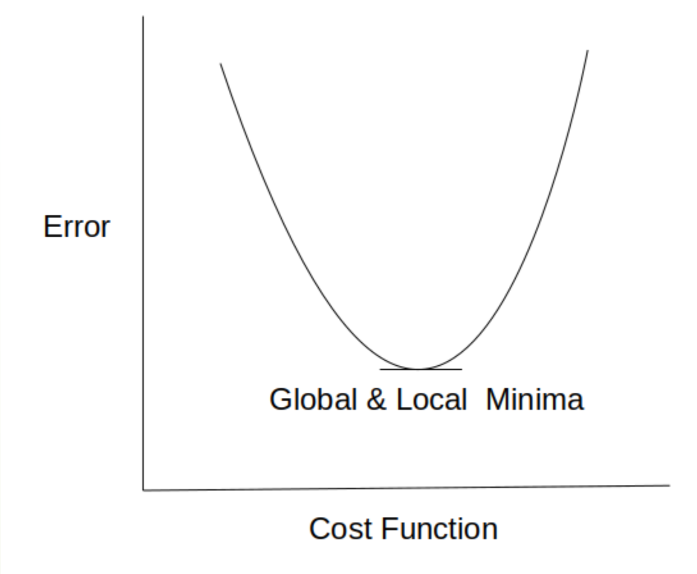


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \dots)$$

Cost Function

- In linear regression, we use the Mean squared error which was the difference between y_predicted and y_actual and this is derived from the maximum likelihood estimator.
- The graph of the cost function in linear regression is like this:

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



Cost Function in Logistic Regression

- In logistic regression $\hat{Y}$ is a non-linear function ($\hat{Y}=1/1+ e^{-z}$).
- If we use this in the above MSE equation, then it will give a non-convex graph with many local minima as shown:

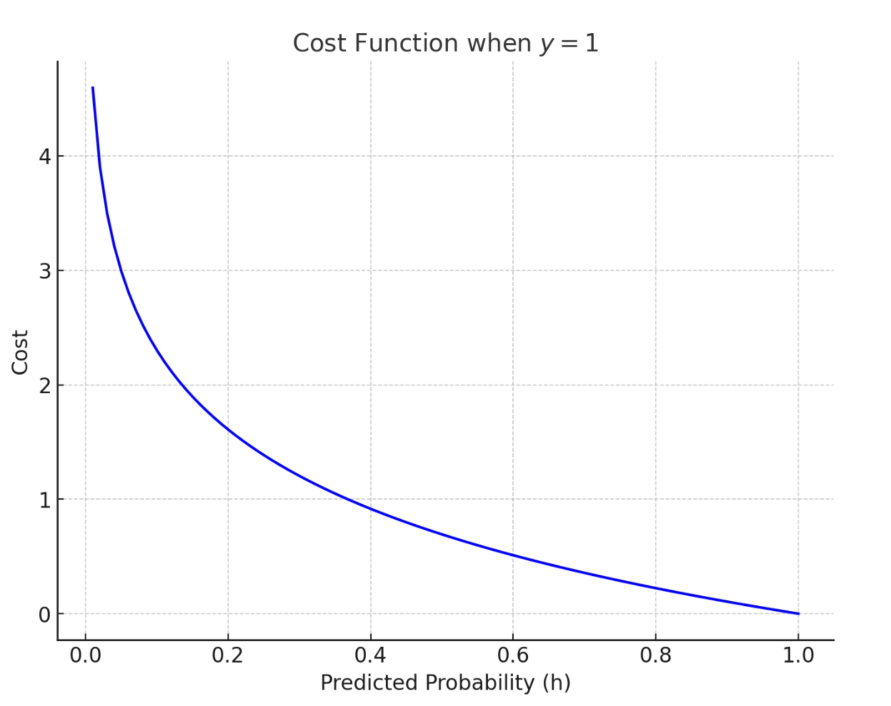
Cost Function in Logistic Regression

- The problem here is that this cost function will give results with local minima, which is a big problem because then we'll miss out on our global minima and our error will increase.
- In order to solve this problem, we derive a different cost function for logistic regression called log loss which is also derived from the maximum likelihood estimation method.

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

Logistic Regression Cost Function

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



Cost $= 0$ if $y = 1, h_\theta(x) = 1$

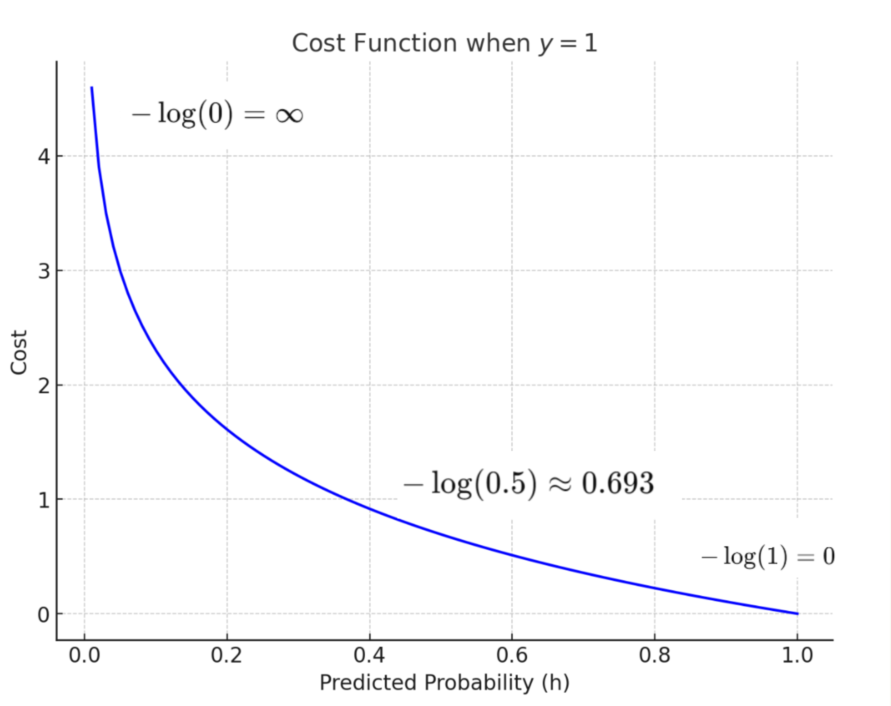
But as $h_\theta(x) \rightarrow 0$

$Cost \rightarrow \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

Logistic Regression Cost Function

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



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

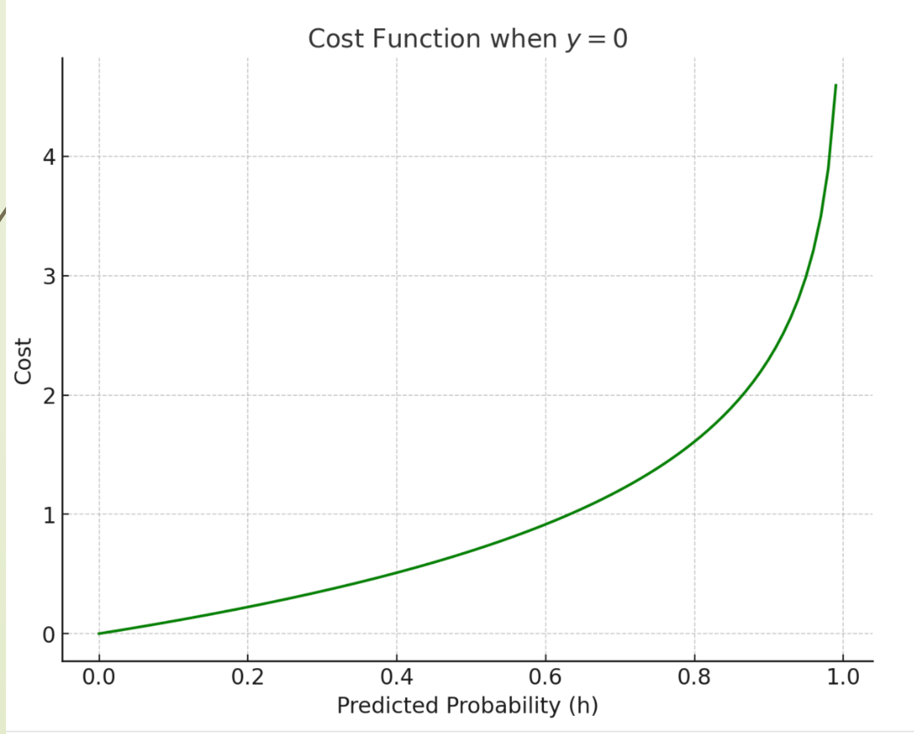
But as $h_\theta(x) \to 0$

$Cost \to \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x;\theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

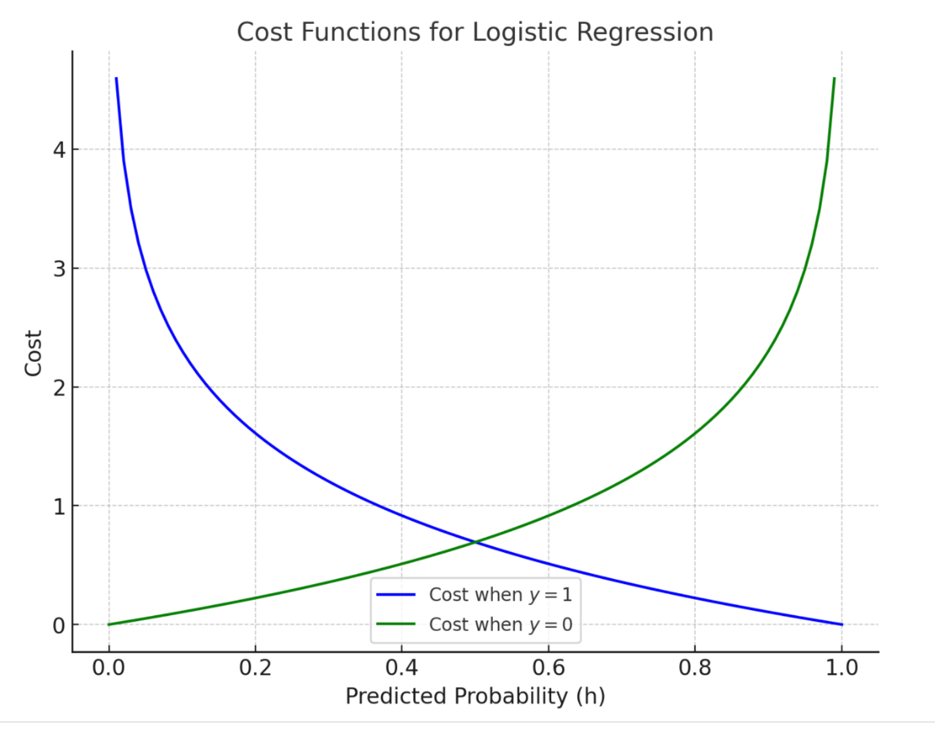
Logistic Regression Cost Function

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



Logistic Regression Cost Function

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



Logistic Regression Cost Function

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

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

Note: $y = 0$ or 1 always

Logistic Regression Cost Function

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

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

To fit parameters θ :

$$\min_\theta J(\theta)$$

To make a prediction given new x :

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

Logistic Regression Cost Function

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

Want $\min_\theta J(\theta)$:

Repeat$\{$

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

$\}$ (simultaneously update all θ_j)

Gradient Descent

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

Want $\min_\theta J(\theta)$:

Repeat $\{$

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

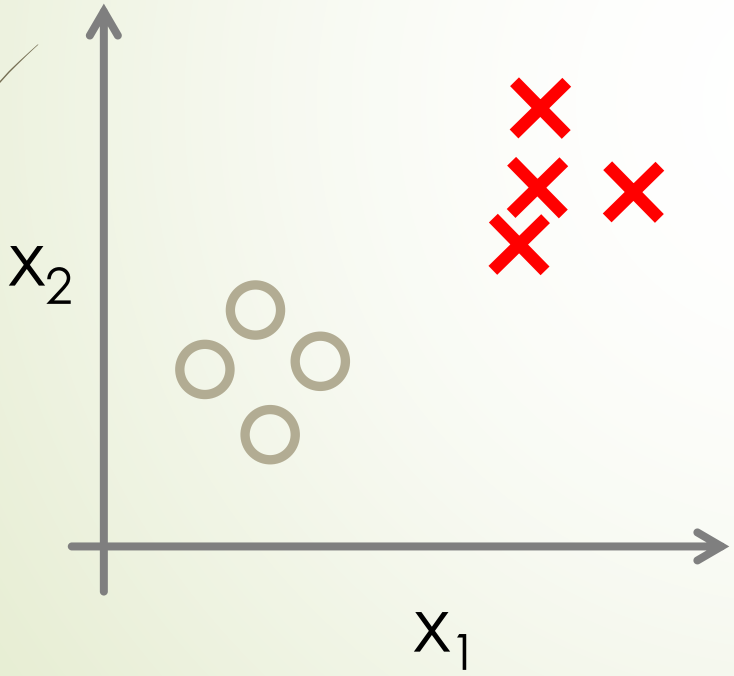
(simultaneously update all θ_j)

$\}$

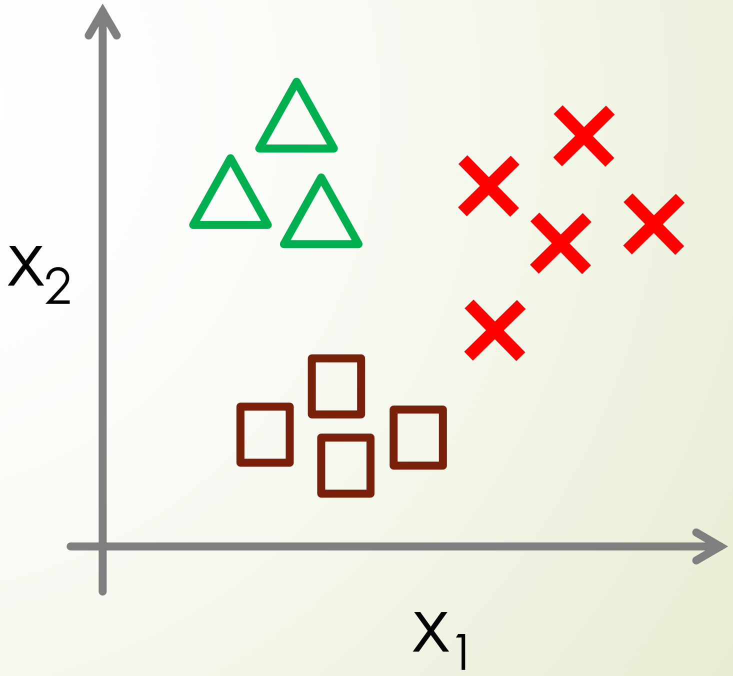
Algorithm looks identical to linear regression!

Multi-class classification:

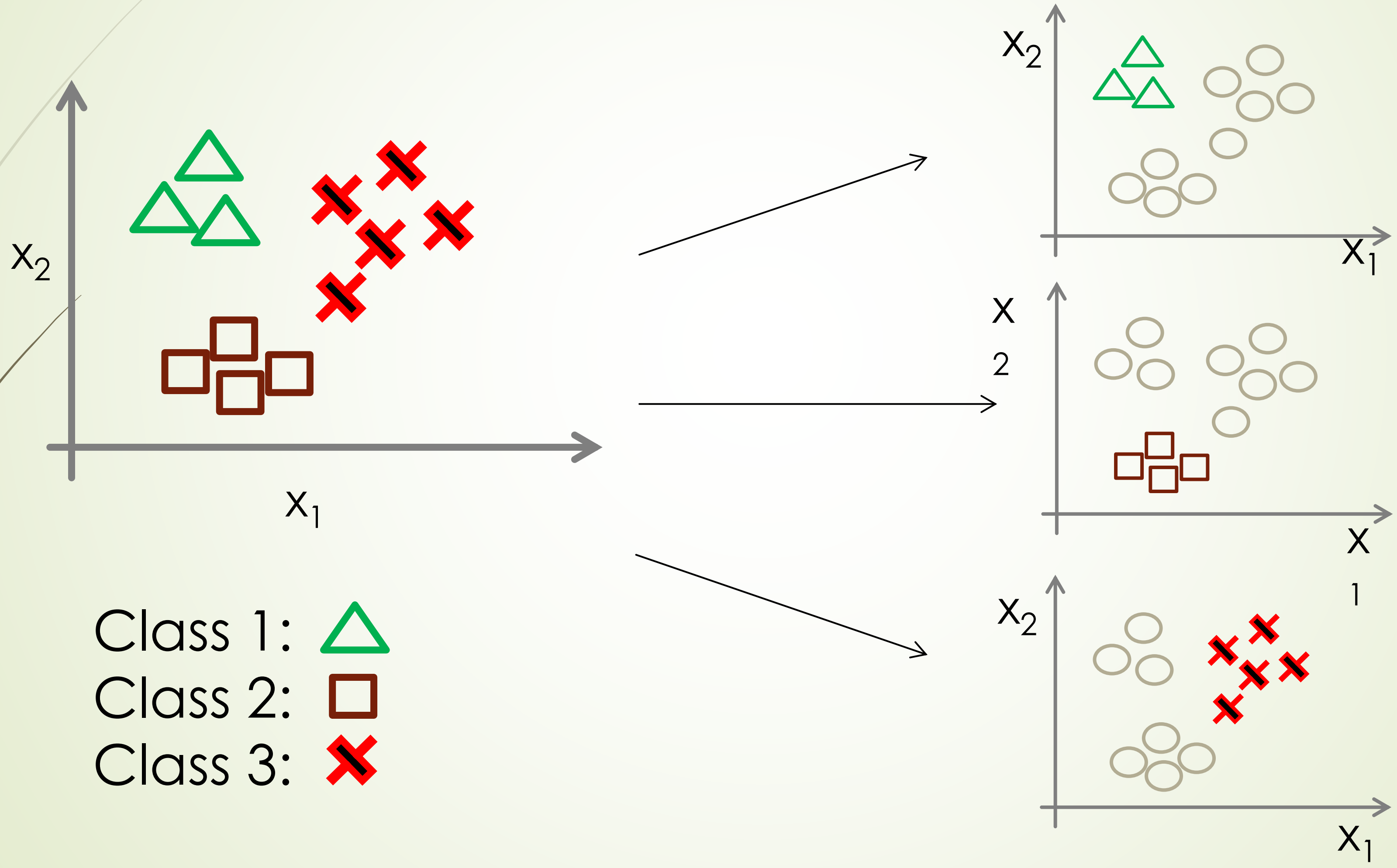
Binary classification:



Multi-class classification:



One-vs-all (one-vs-rest):



Comparison Between Linear and Logistic Regression

Linear Regression	Logistic Regression
Predicts a **continuous** dependent variable using independent variables.	Predicts a **categorical** (binary) dependent variable using independent variables.
Solves **regression** problems.	Solves **classification** problems.
Uses the **Least Squares** estimation method for accuracy.	Uses the **Maximum Likelihood** estimation method for accuracy
Fits a **straight line** (best-fit line) to predict the output.	Fits an **S-curve** (logistic function) to classify the samples.
Requires a **linear relationship** between the dependent and independent variables.	Does **not require** a linear relationship between dependent and independent variables.

References

- https://realpython.com/logistic-regression-python/
- https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/