Logistic Regression

Logistic Regression

- As the name suggests it is an algorithm based on regression
- But we need to discuss about the types of data we can have
- Are the following data inputs equal?
  - Height of a person — 1.92, 1.8, 1.59…
  - Number of students in class — 20, 6, 24…
  - Gender — Male, Female…
  - Grade — A, C, F…
  - GPA — 3.89, 3.22, 1.85…
  - Degree name — BFA, RTIS…
  - Economic status — Low, Medium, High

Data Types

Height of a person: 1.92, 1.8, 1.59…

Number of students in class: 20, 6, 24…

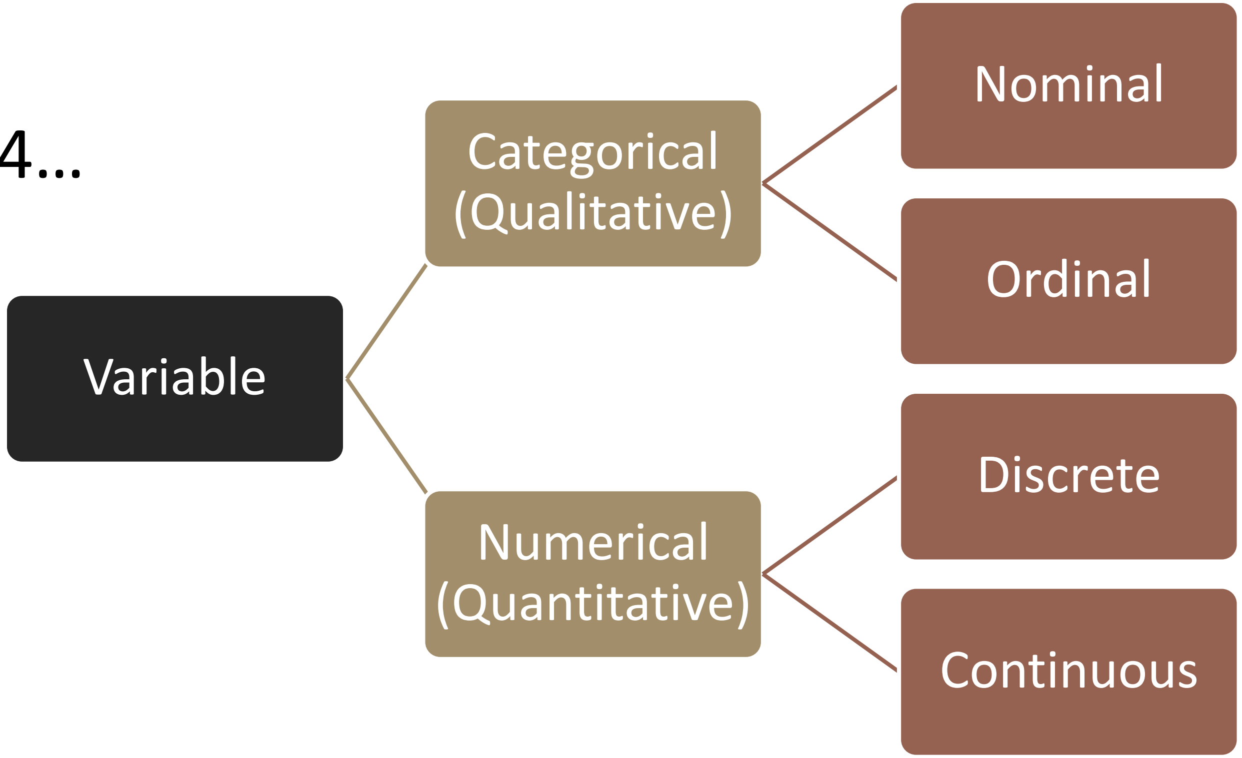
Gender: Male, Female…

Grade: A, C, F…

GPA: 3.89, 3.22, 1.85…

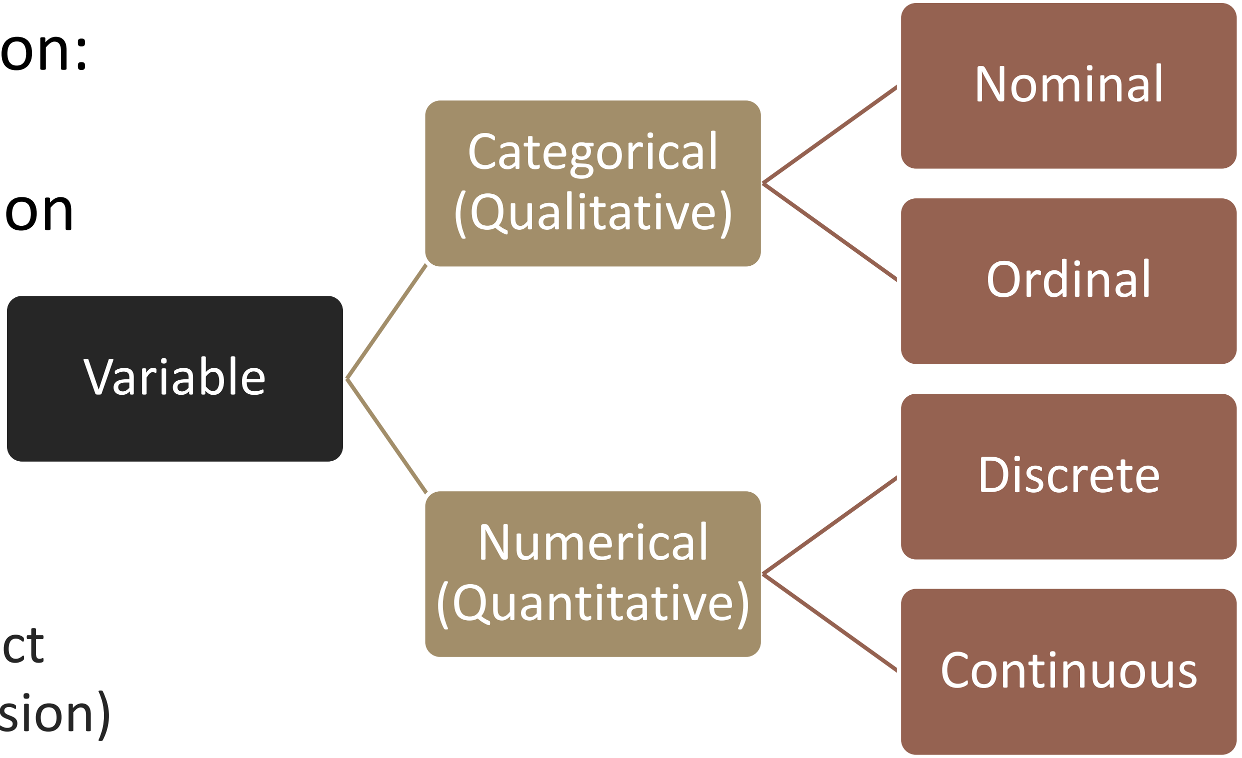
Degree name: BFA, RTIS…

Economic status: Low, Medium, High



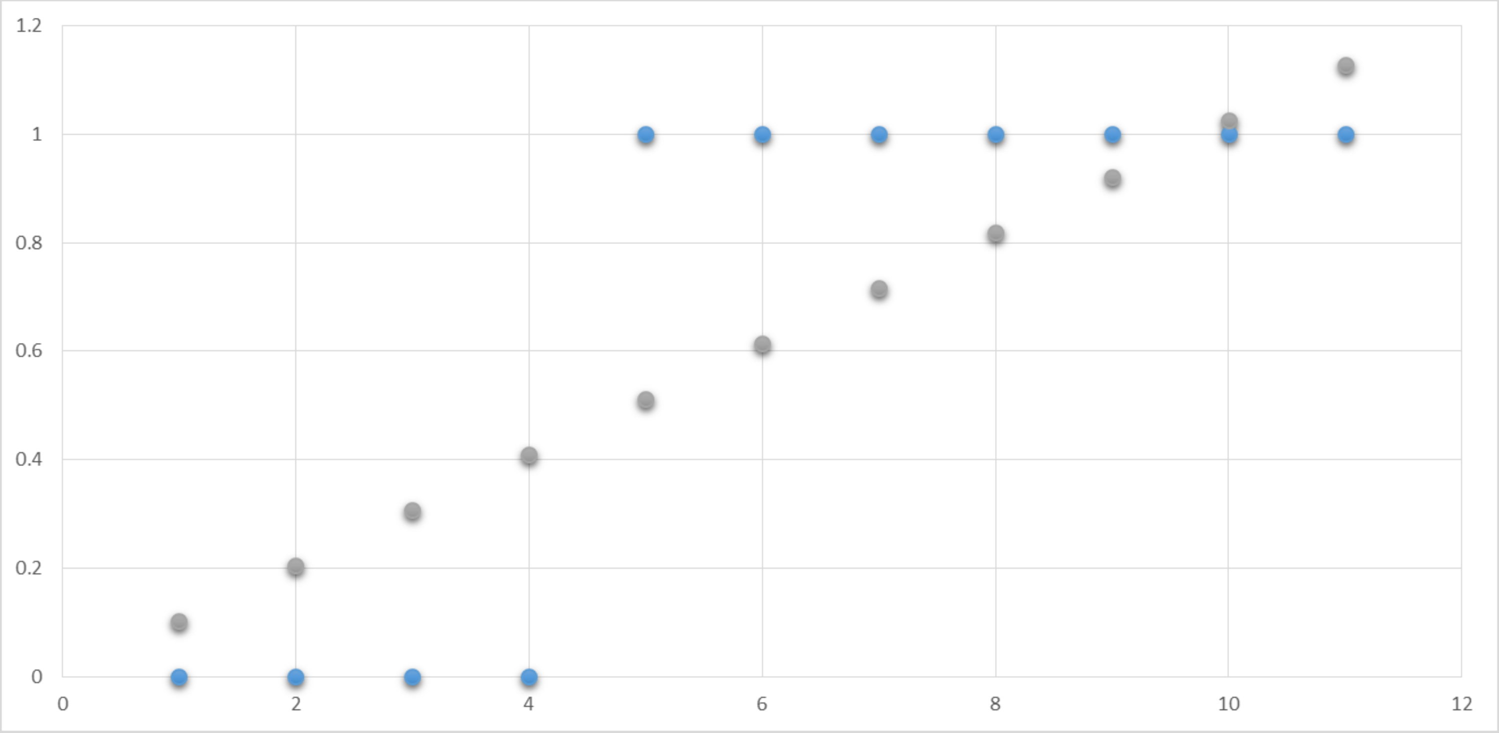
Logistic Regression

- Linear Regression was mainly focused on:
  - Numerical continuous
- For Logistic Regression a clear distinction between outcomes is needed:
  - Categorical outcome
    - Product: Functional/Defective
    - Email: Spam/Not Spam
    - Student application: Accepted/Rejected
    - Customer: Will/Won't buy a specific product
    - Output value y ∈ {0, 1}, where (Binary decision)
      - 0 = Negative output
      - 1 = Positive output



Linear Regression…

Input	0	1	2	3	4	5	6	7	8	9	10	11
Output	0	0	0	0	0	1	1	1	1	1	1	1



Logistic Regression

- Linear regression is not a good approach for such problems
  - Might be able to separate a small number of samples, but will struggle as samples’ range increases
  - Need to separate samples
    - As opposed to generating an equation that passes through them

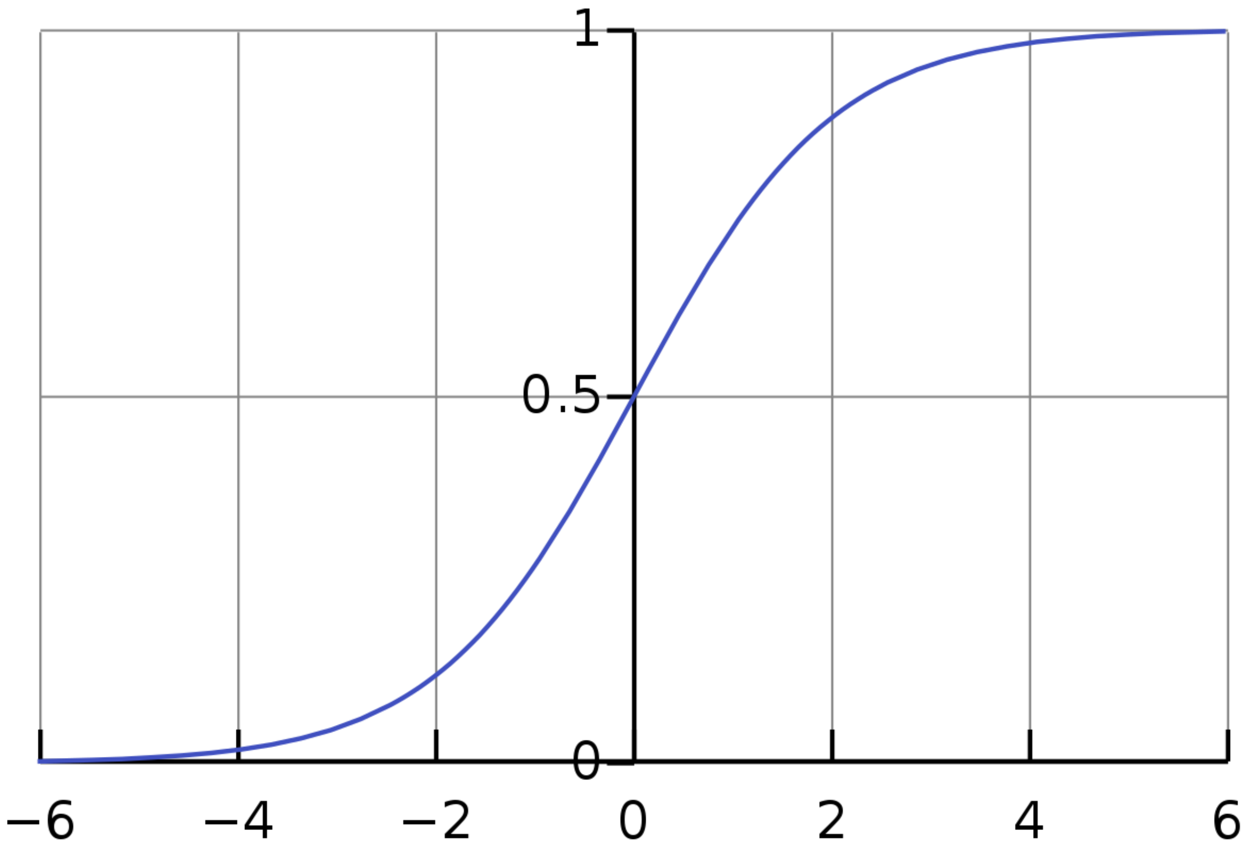
Logistic Function

- Referred to as Sigmoid function

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

- σ(z) generates a value between 0 and 1
  - $\sigma(z) > 0.5$ if $z > 0$
  - $\sigma(z) < 0.5$ if $z < 0$
- Apply the sigmoid function to the hypothesis equation:
  - $h_\theta(x) = \theta^T x$ becomes $h_\theta(x) = \sigma(\theta^T x)$
- What is the shape of Sigmoid function?
  - Range?

Sigmoid Function



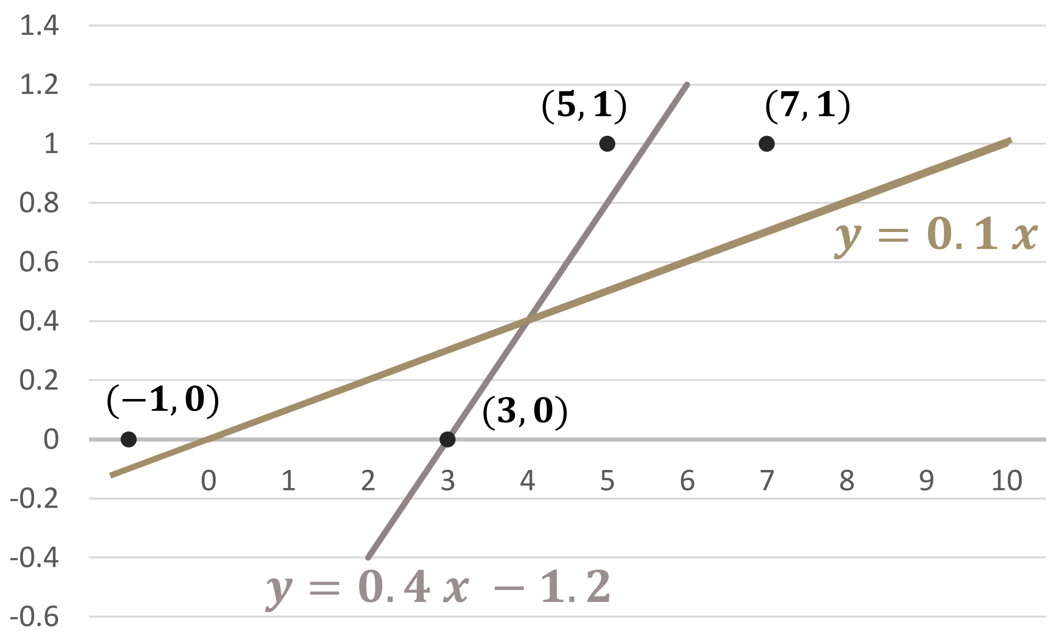
Logistic Function

$$y = 0.1\,x$$

$$(-1,0) \rightarrow y = 0.1 \; * -1 = -0.1$$
$$(3,0) \rightarrow y = 0.1 \; * 3 = 0.3$$
$$(5,1) \rightarrow y = 0.1 \; * 5 = 0.5$$
$$(7,1) \rightarrow y = 0.1 \; * 7 = 0.7$$

$$y = 0.4\,x \; - 1.2$$

$$(-1,0) \rightarrow y = 0.4 \; * -1 - 1.2 = -1.6$$
$$(3,0) \rightarrow y = 0.4 \; * 3 - 1.2 = 0$$
$$(5,1) \rightarrow y = 0.4 \; * 5 - 1.2 = 0.8$$
$$(7,1) \rightarrow y = 0.4 \; * 7 - 1.2 = 1.6$$



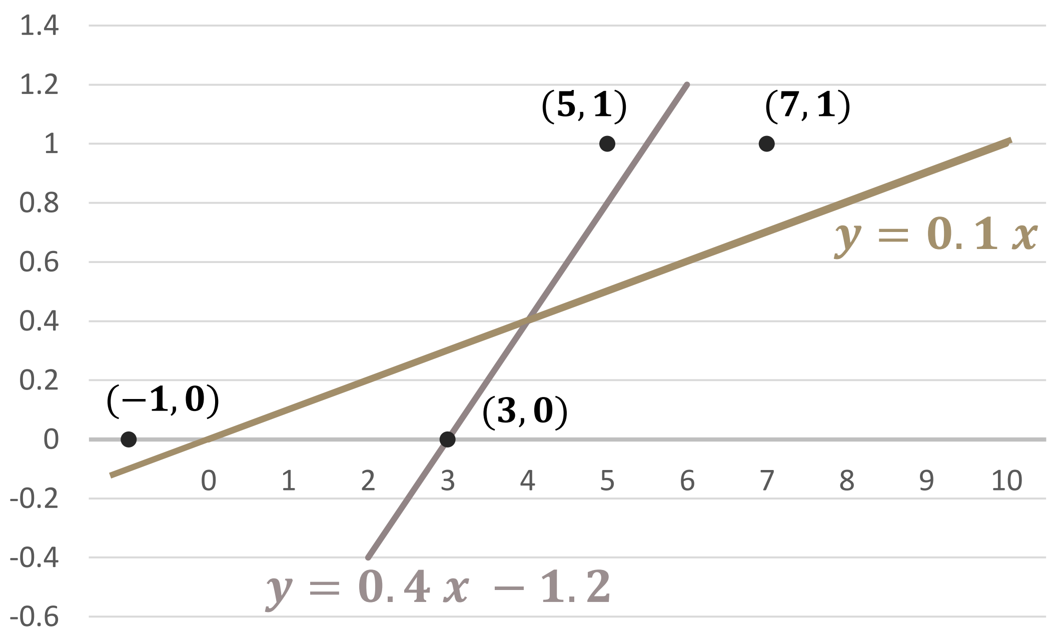
Logistic Function

$$y = 0.1\,x$$

$$(-1,0) \rightarrow h_\theta(-1) = \sigma(-0.1) = 0.475$$
$$(3,0) \rightarrow h_\theta(3) = \sigma(0.3) = 0.574$$
$$(5,1) \rightarrow h_\theta(5) = \sigma(0.5) = 0.622$$
$$(7,1) \rightarrow h_\theta(7) = \sigma(0.7) = 0.668$$

$$y = 0.4\,x - 1.2$$

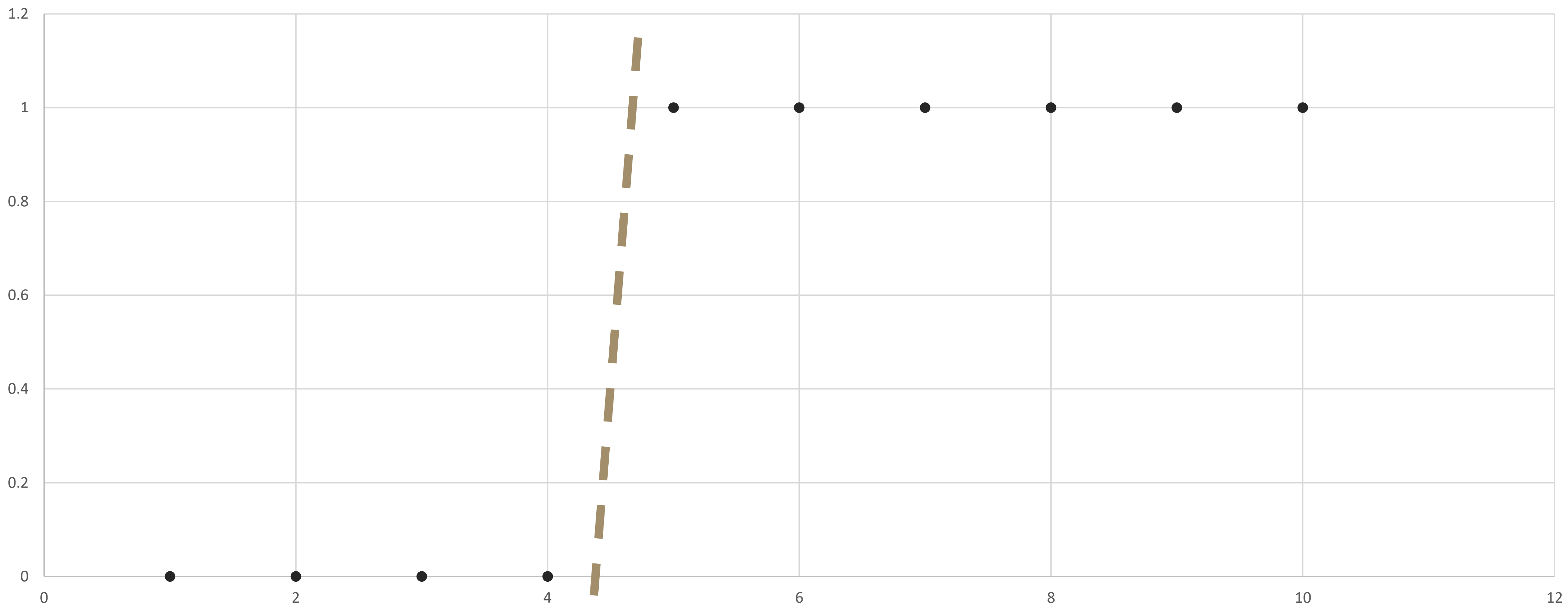
$$(-1,0) \rightarrow h_\theta(-1) = \sigma(-1.6) = 0.168$$
$$(3,0) \rightarrow h_\theta(3) = \sigma(0) = 0.5$$
$$(5,1) \rightarrow h_\theta(5) = \sigma(0.8) = 0.69$$
$$(7,1) \rightarrow h_\theta(7) = \sigma(1.6) = 0.832$$



Logistic Function

- Example: Determining if a product is defective (1 means defective):
  - $h_\theta(x) = \sigma(\theta^T x) = 0.75$
  - Predicted y = 1
  - 75% confident that the product is defective
- Basically the hypothesis function will estimate the probability of the input being categorized as y=1

Decision Boundary



Prediction & Decision Boundary

- $\sigma(\theta^T x)$ is now the output (prediction)
- $\theta^T x=0$ now represents the **decision boundary**
- $y = 1$ when $\theta_0 + \theta_1 x_1 + \theta_2 x_2 \geq 0$ $(x_0 = 1)$
- $y = 0$ when $\theta_0 + \theta_1 x_1 + \theta_2 x_2 < 0$ $(x_0 = 1)$

Prediction & Decision Boundary

- $\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$ separates the "1" samples from the "0" samples
- Examples:
  - Line
  - Circle
  - Ellipse

Decision Boundary

- Assume the following decision boundary: $-4 + x_1^2 = 0$
- What is the predicted value for $x_1 = 0$, $x_1 = 2$ and $x_1 = -10$?
  - $x_1 = 0$: $h_\theta(x) \approx 0.179$
  - $x_1 = 2$: $h_\theta(x) \approx 0.5$
  - $x_1 = -10$: $h_\theta(x) \approx 1$

- What if the boundary is $-9 + x_1^2 = 0$ and test same x_1 values?
  - $x_1 = 0$: $h_\theta(x) \approx 0$
  - $x_1 = 2$: $h_\theta(x) \approx 0.01$
  - $x_1 = -10$: $h_\theta(x) \approx 1$

Cost

- Using the cost function from Linear Regression:

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

- Is it exactly the same?
  - $h_\theta(x)$ is now a sigmoid function
  - $h_\theta(x) = \sigma(\theta^T x)$ (Not simply $h_\theta(x) = \theta^T x$)
- Using that same cost function results in a non-convex curve
  - What are the consequences of this?

Cost

- A non-convex function has several local optima
  - Gradient descent won't necessarily converge to the global minimum
- The reason is that $h_\theta(x)$ is very non-linear, since it's now a Sigmoid function
  - Note: $h_\theta(x)$ in linear regression could also be non-linear

Cost Function

- What we want:
  - When y = 1:
    - Error is very high if predicted $h_\theta(x)$ is closer to 0
    - Error is reduced as predicted $h_\theta(x)$ gets closer to 1
  - When y = 0:
    - Error is very high if predicted $h_\theta(x)$ is closer to 1
    - Error is reduced as predicted $h_\theta(x)$ gets closer to 0

Cost

- Approach for converting the cost function to become linear:

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

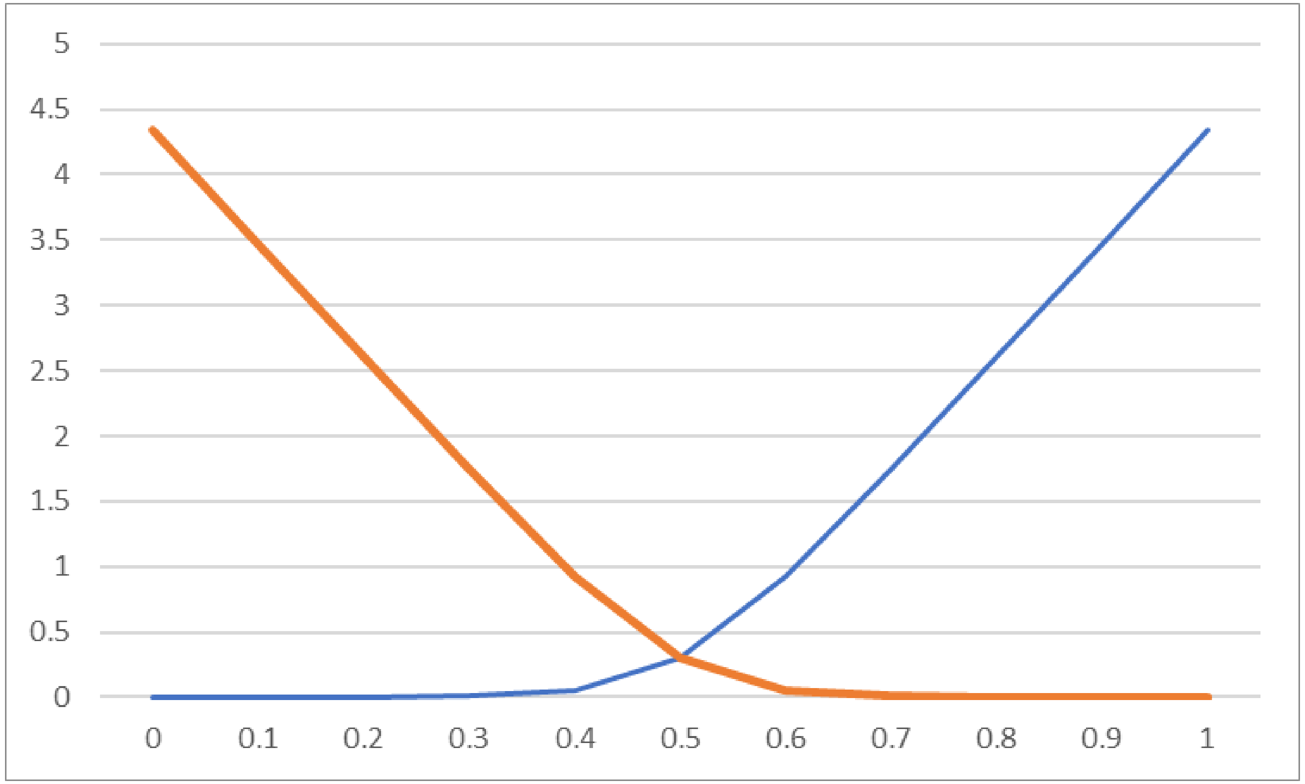
- The "Error" (and therefore the Cost function) are different when y = 1 or y = 0

$$\text{y = 1: } Error(h_\theta(x), y) = -\log(h_\theta(x))$$
$$\text{y = 0: } Error(h_\theta(x), y) = -\log(1 - h_\theta(x))$$

- How do those two graphs look like?

Cost

$y = -log(h_\theta(x))$



$y = -log(1 - h_\theta(x))$

Cost

$$Error\left(h_\theta\left(x^i\right), y^i\right) = -y \cdot \log\left(h_\theta(x)\right) - (1-y) \cdot \log(1 - h_\theta(x))$$

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

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

Works for y = 0 and y = 1

Gradient Descent

- Similar to linear regression
- Based on deriving the cost equation

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

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

Gradient Descent

- Derivation looks similar to the one from linear regression
  - But the **Hypothesis function is different**!
  - $h_\theta(x)$ is different. It's now a Sigmoid function.
  - $h_\theta(x) = \theta^T x$ is now $h_\theta(x) = \sigma(\theta^T x)$

Regularization

- Same issue as with Linear Regression
- Overfitting could occur
- Add regularization component
  - Pushes θ's values to remain small

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

Regularization

- Deriving the cost equation yields the same function from Linear Regression
  - Reminder: Hypothesis function is different!
- Every iteration, each θ is decreased by the cost function derivative
- Similarly to linear regression, θ_0 is not regularized

Regularization

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

- Same as in linear regression, but $h_\theta(x)$ is different

Gradient Descent Algorithm

- Pick features (x_1, x_2, etc.)
- Pick a value for the learning rate α, and regularization term λ
- Set the equation's complexity
  - Example: $y = h_\theta(x) = \sigma(\theta^T x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2$, $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2^2)$
- For each θ_j, where $0 < j < n$:
  - Compute the cost function derivative with respect to θ_j
  - Adjust θ_j by – α*Cost Derivative
    - Don't update θ_j's value yet
    - Save to a temporary location
- For each θ_j:
  - Update θ_j's value from the temporary location

References

- Notes by Antoine Abi Chacra, DigiPen Institute of Technology
- Wikipedia