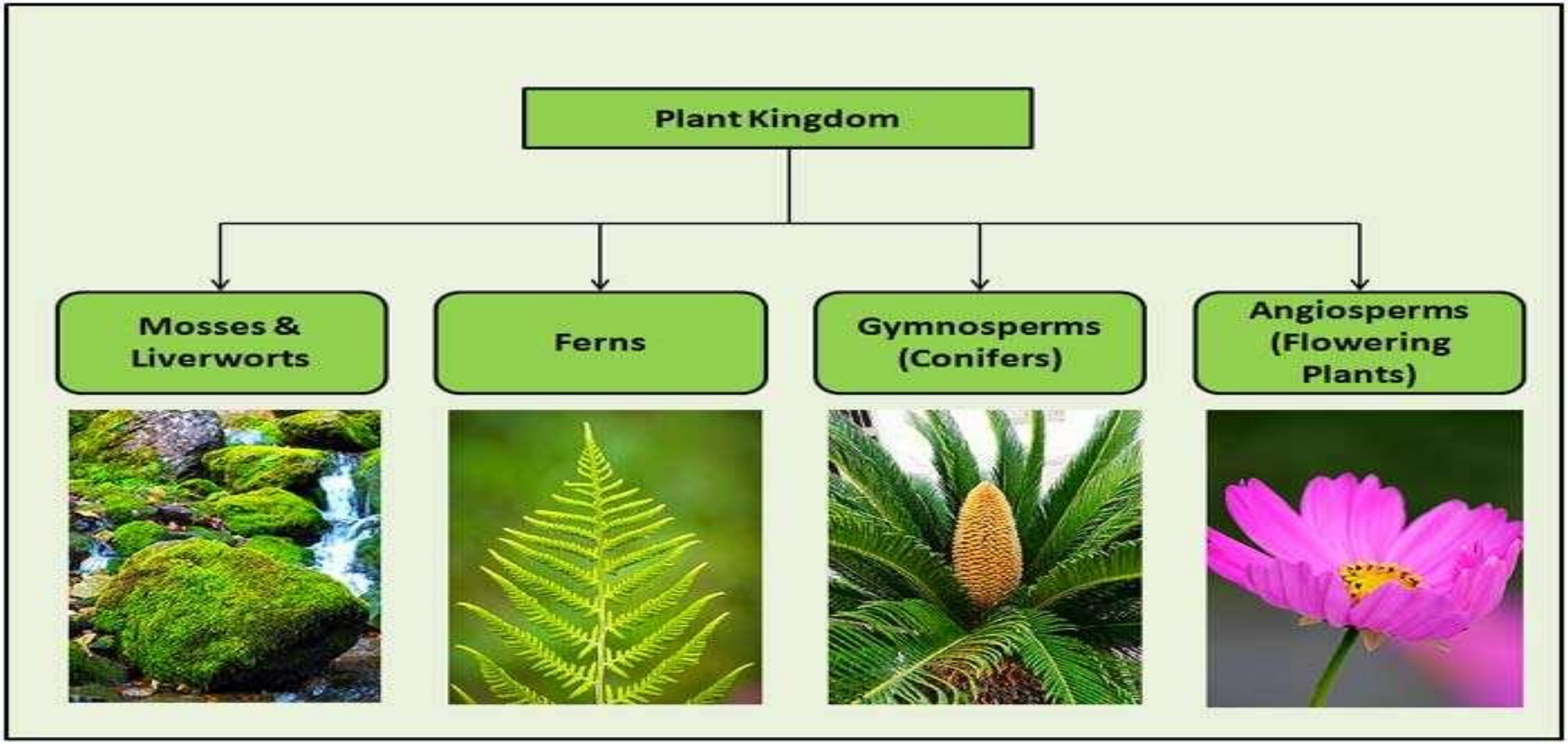
Logistic Regression

Guowei Wei
Department of Mathematics
Michigan State University

References:
Duc D. Nguyen's lecture notes
Andrew Ng's notes
Wikipedia

Introduction

Statistical classification: identifying to which of a set of categories a new observation belongs, on the basis of a training set of data.



Introduction

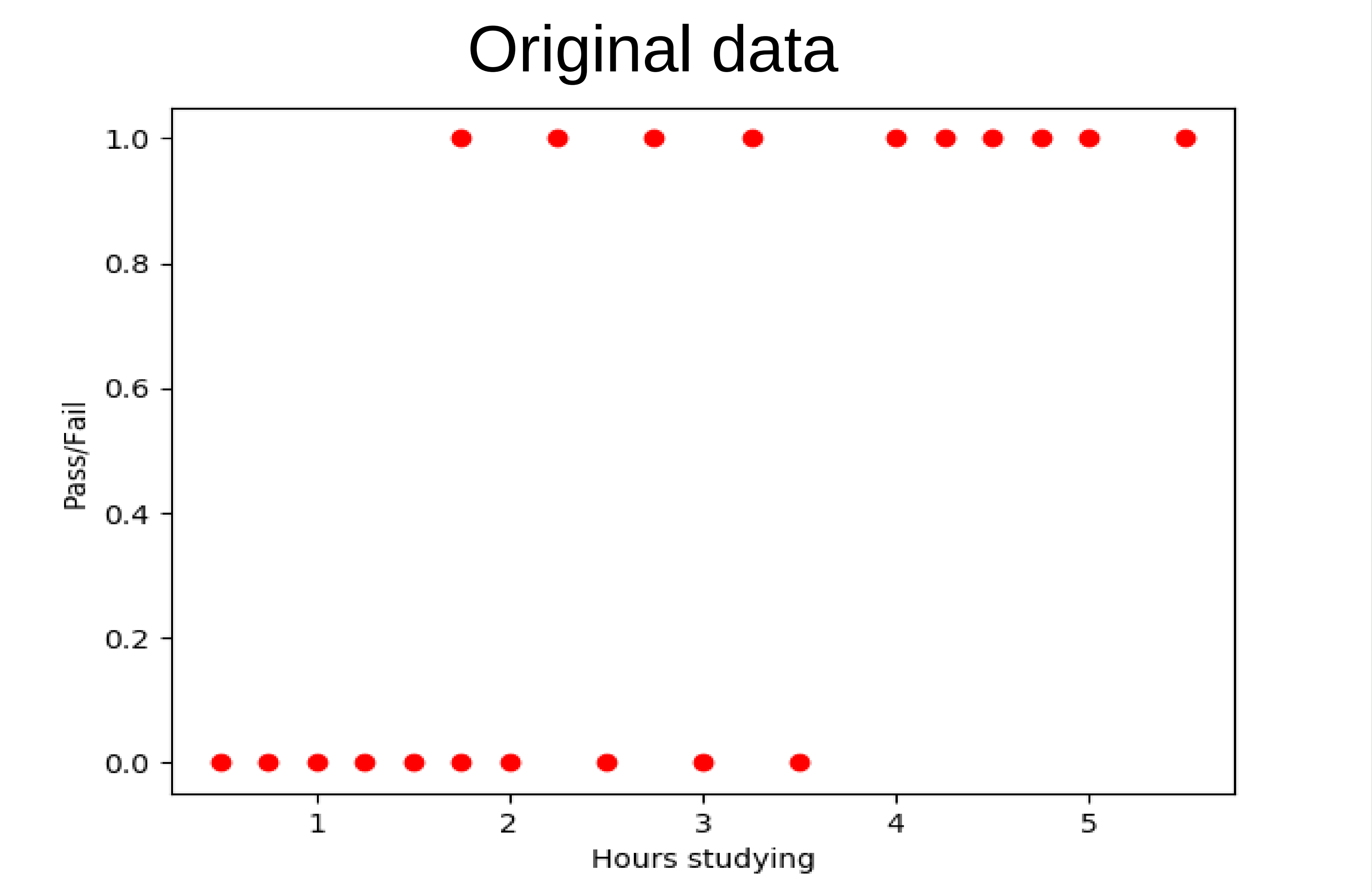
- Classification
- Examples:
  - Matter: Toxic / Not Toxic ?
  - Students: Pass / Fail ?
  - Van Gogh Painting: Real / Fake?
- Labels $y \in \{0,1\}$. 0: Negative class (Fail), 1: Positive class (Pass)
- We can have more than two labels $y \in \{0,1,2,3\}$ for DNA (cytosine [C], guanine [G], adenine [A] and thymine [T])

Example

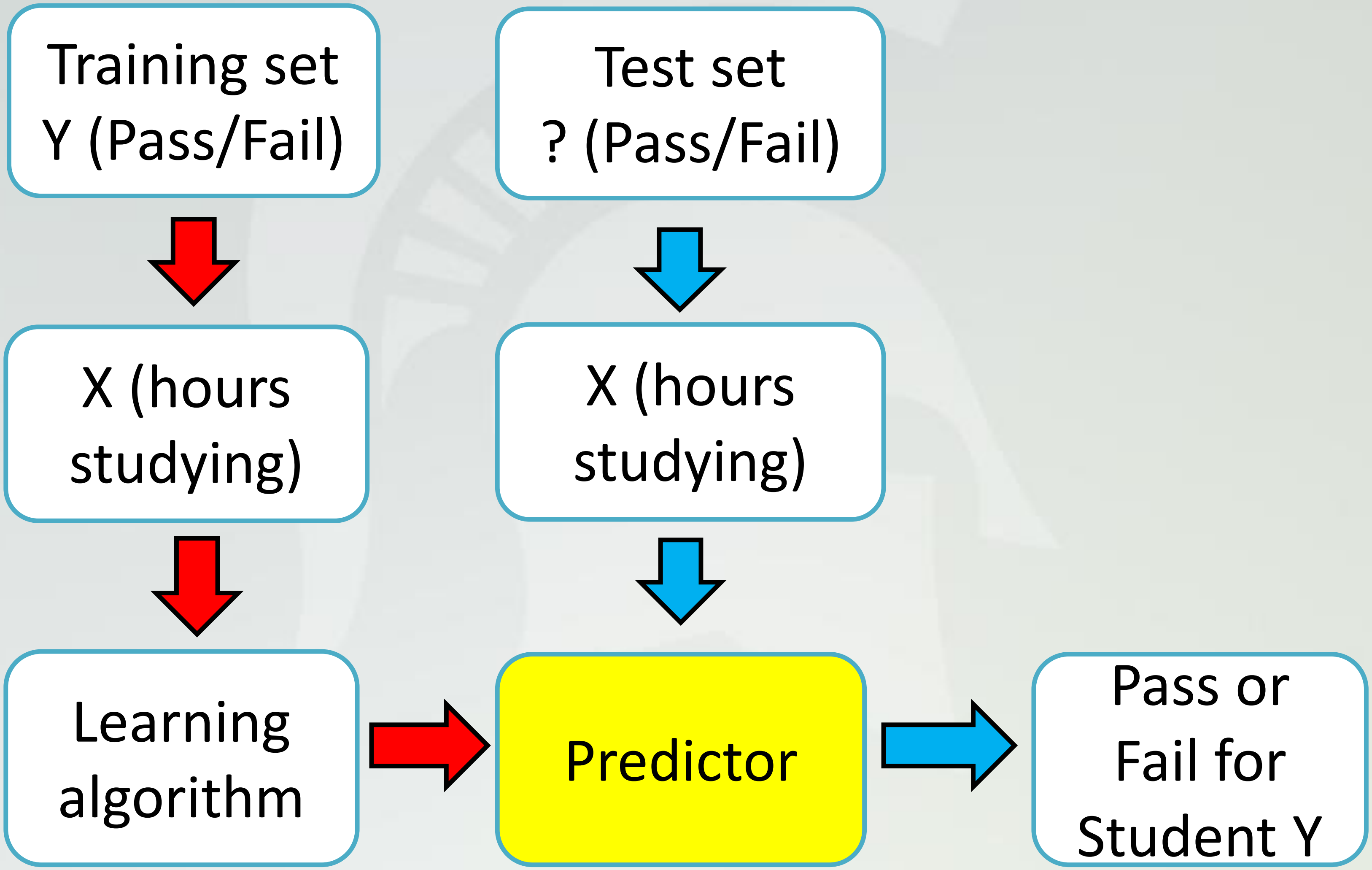
- Performance of group of 20 students spend between 0 and 6 hours studying for an exam

Hours	Pass/Fail
0.5	Fail
0.75	Fail
1.75	Pass
2.25	Pass
…	…

Example


Original data
1.0
0.8
0.6
0.4
0.2
0.0
Pass/Fail
1
2
3
4
5
Hours studying

Model Representation



Predictor Construction

- Wish to construct a predictor:

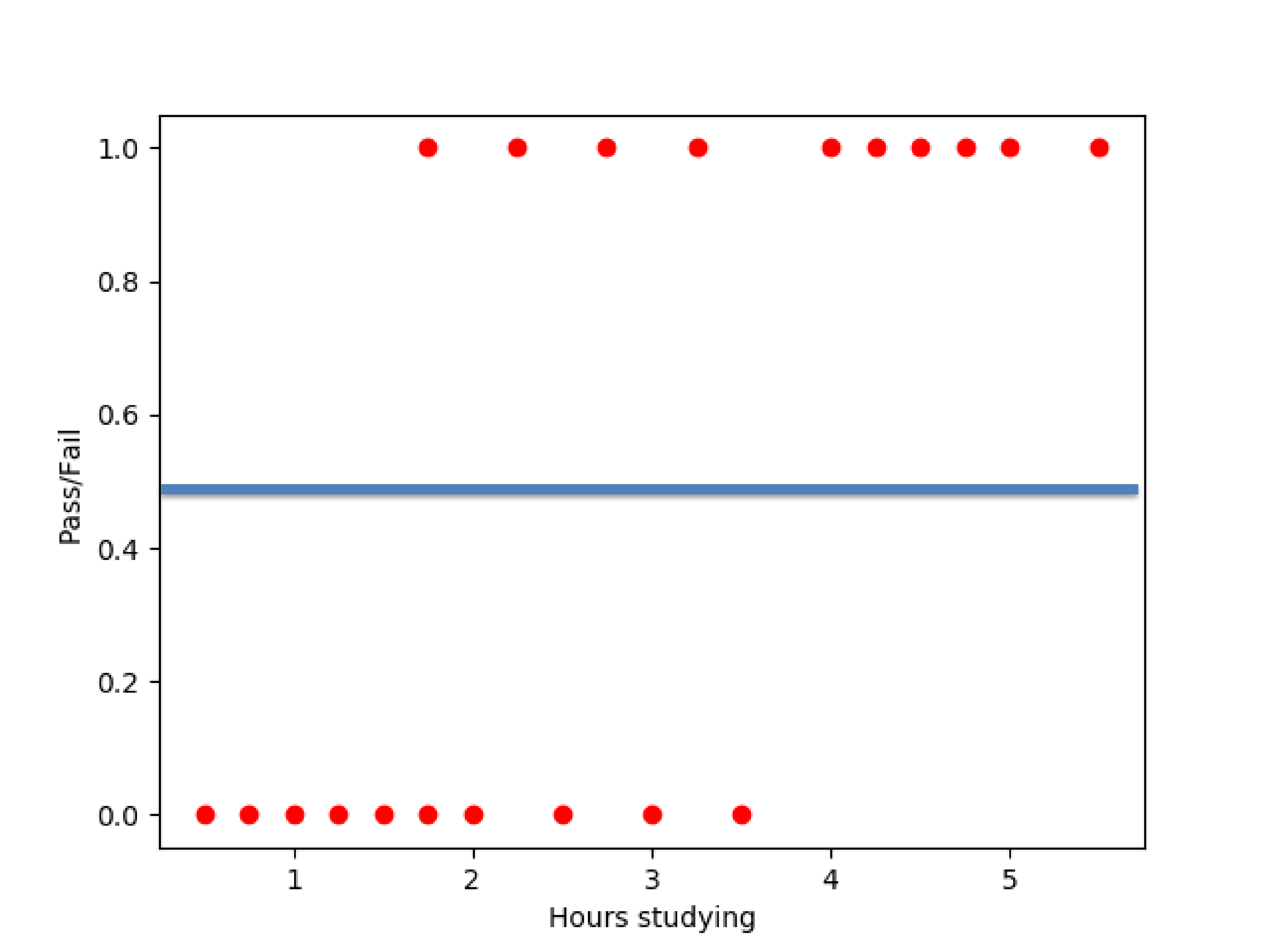
$$p_{\mathbf{c}}(\mathbf{x}) = ?$$

- Our labels only has two values 0 and 1
- Predictor always give a real value.
- $0 \leq p_{\mathbf{c}}(\mathbf{x}) \leq 1$
- To get classification, choose a threshold z:

$$p_{\mathbf{c}}(\mathbf{x}) < z \text{ then } y = 0$$
$$p_{\mathbf{c}}(\mathbf{x}) \geq z \text{ then } y = 1$$

- With two labels $\{0,1\}$, it is natural to choose $z = 0.5$

Predictor Construction



Predictor Construction

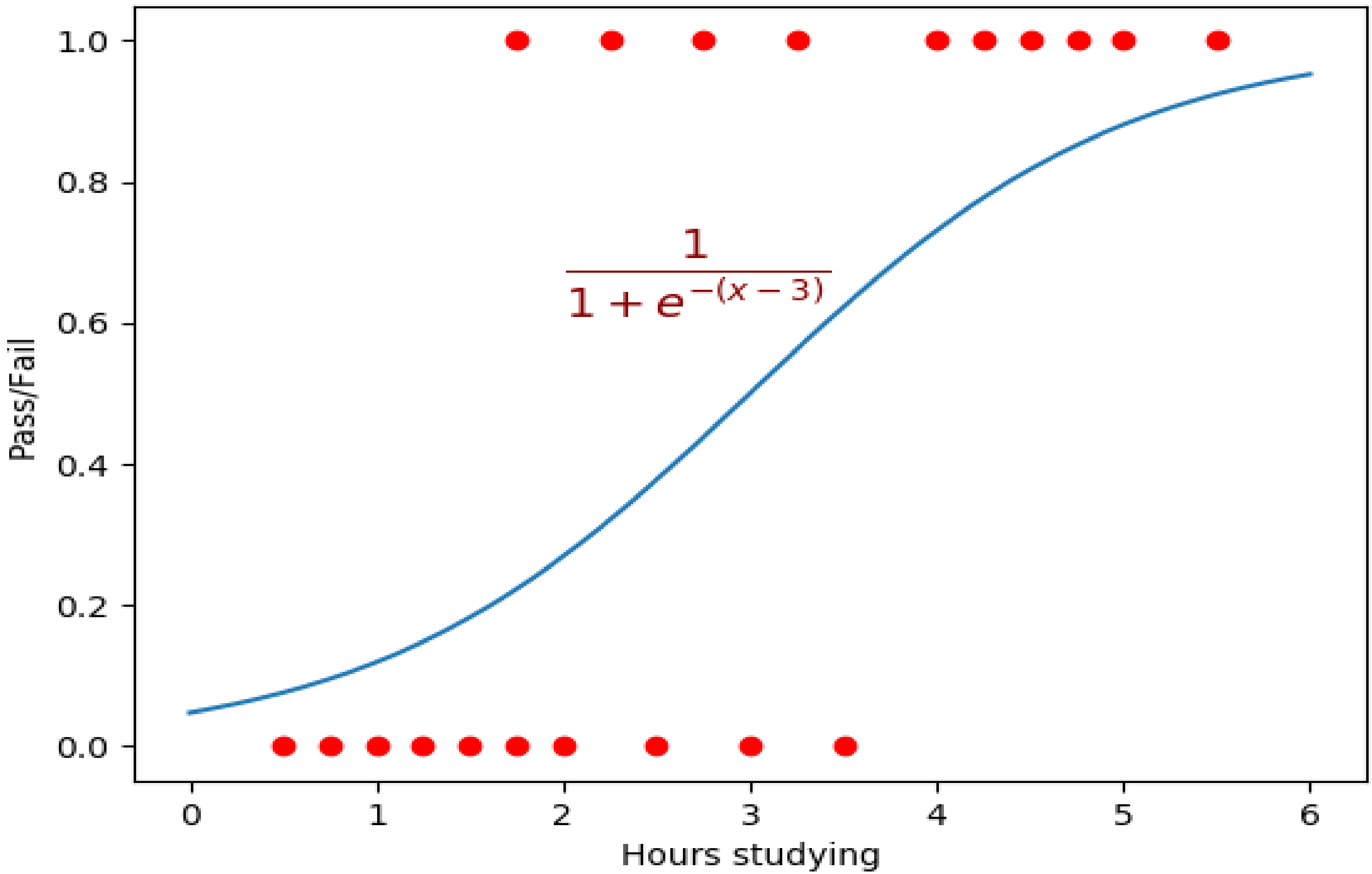
- We choose $\boxed{\text{Sigmoid / Logistic function:}}$

$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{c}^T \mathbf{x}}}$$

(Is this model still linear?)

- If dataset has one feature, i.e., $\mathbf{x} = (1, x_1)^T$, then $\mathbf{c} = (c_0, c_1)^T$

Predictor Construction



Predictor Construction

- If dataset has *n* features, i.e., $\mathbf{x} = (1, x_1, \ldots, x_n)^T$, then $\mathbf{c} = (c_0, c_1, \ldots, c_n)^T$
- Single-layer perceptron

$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1 + e^{-c_0 - c_1 x_1 - c_2 x_2 - \cdots - c_n x_n}}$$

or $$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{c}^T\mathbf{x}}}$$

is the probability that $y = 1$ on input $\mathbf{x}$

Predictor Construction

- We can also use the polynomial logistic function, e.g.,

$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{-c_0-c_1x_1-c_2x_1^2}}$$ or more general:

- $$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{-c_0-c_{11}x_1-c_{12}x_1^2+,\ldots,c_{n1}x_n-c_{n2}x_n^2+,\ldots}}$$

- **Loss function construction:**
  - Linear regression:

$$L(\mathbf{c}) = \sum_{i=1}^{m}\left(p\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Predictor Construction

- We can also use the polynomial logistic function, e.g.,

$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{-c_0-c_1x_1-c_2x_1^2}}$$ or more general:

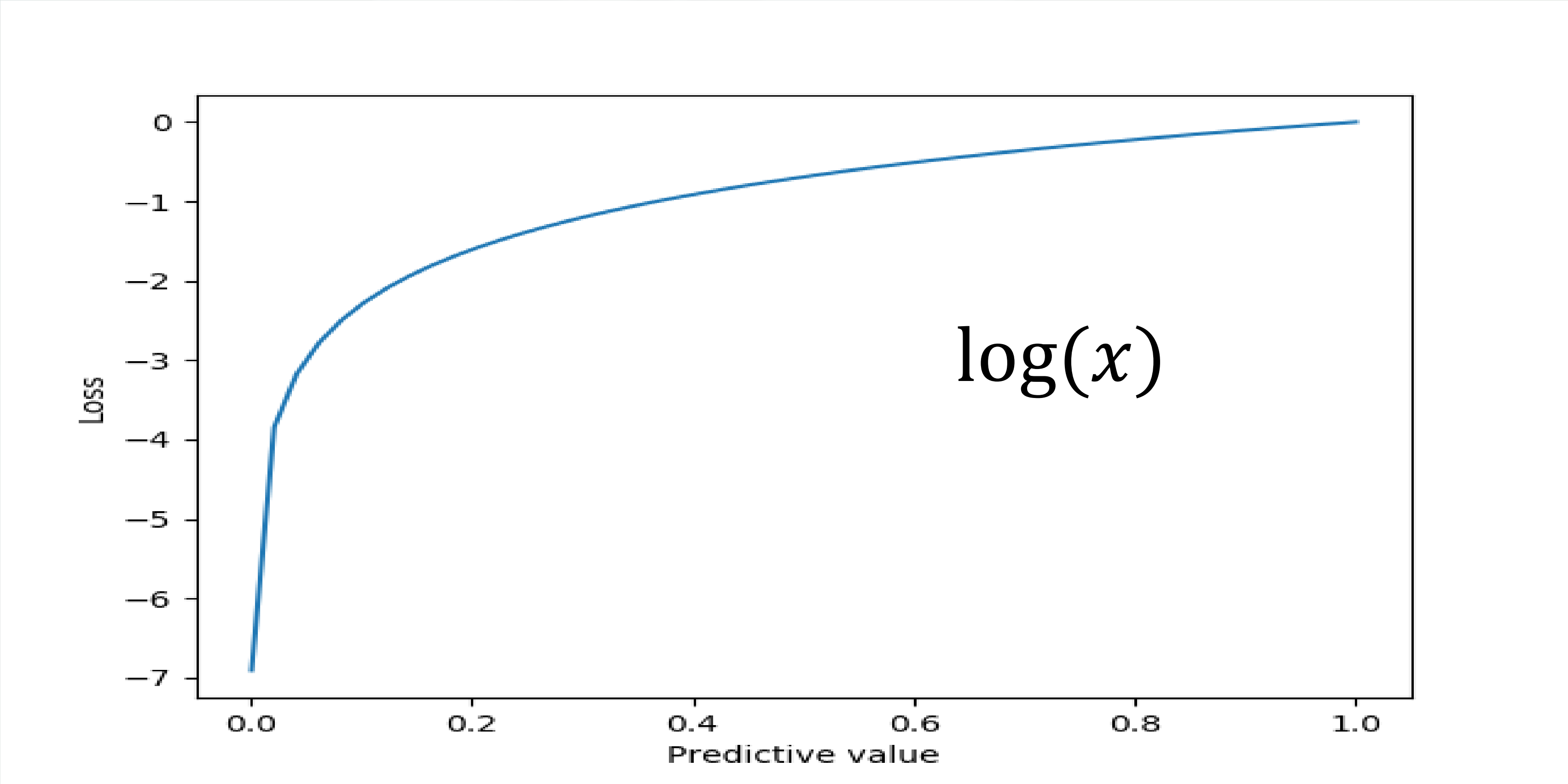
$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{-c_0-c_{11}x_1-c_{12}x_1^2+,\ldots,c_{n1}x_n-c_{n2}x_n^2+,\ldots}}$$

$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{-f(\mathbf{c},\mathbf{X})}}$$

How to determine coefficients?

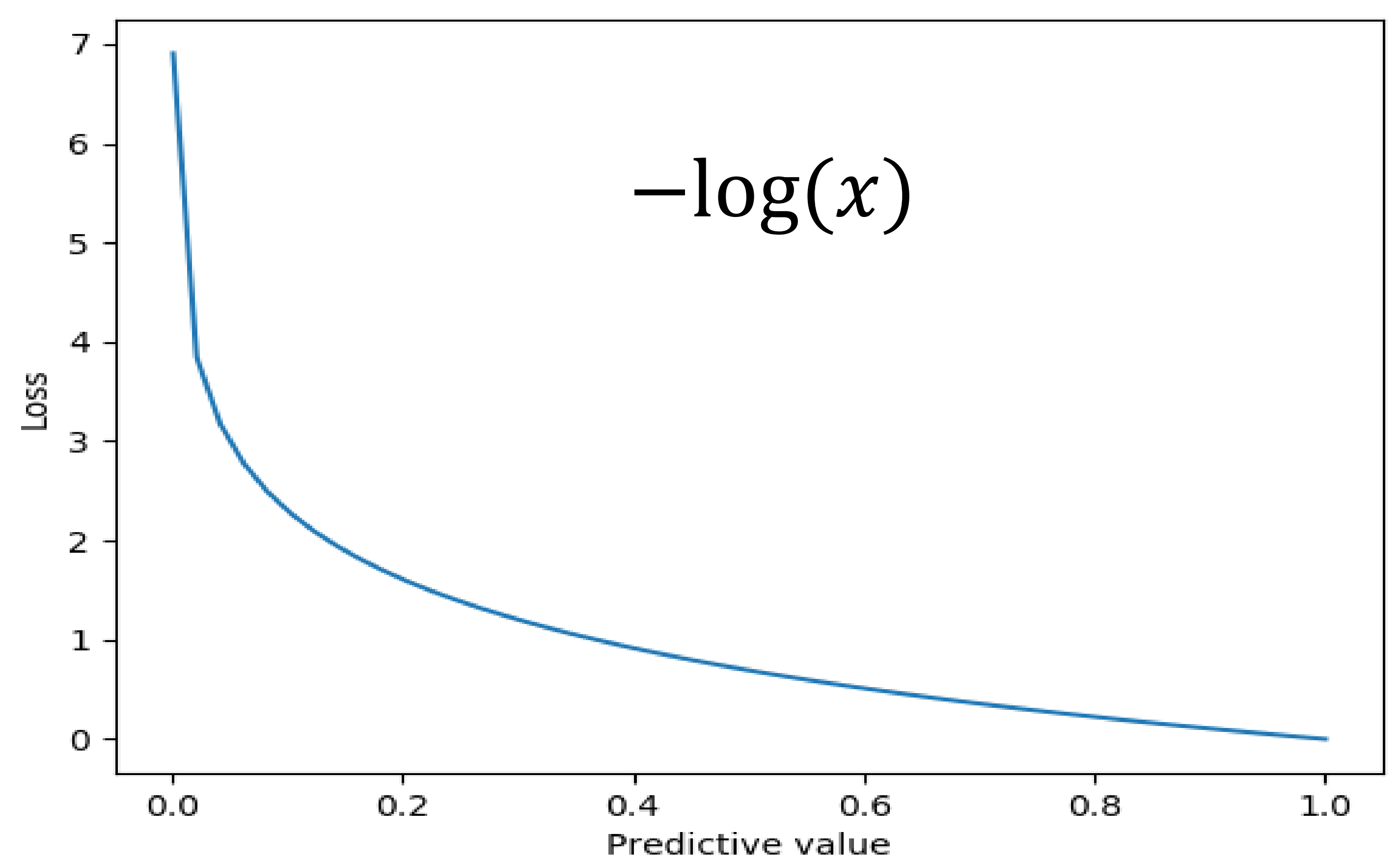
Loss function construction

- When $\boldsymbol{y} = \mathbf{1}$, higher $p_{\mathbf{c}}(\mathbf{x})$ is more accurate ⇒ smaller lost
- Any function reflects that behavior?



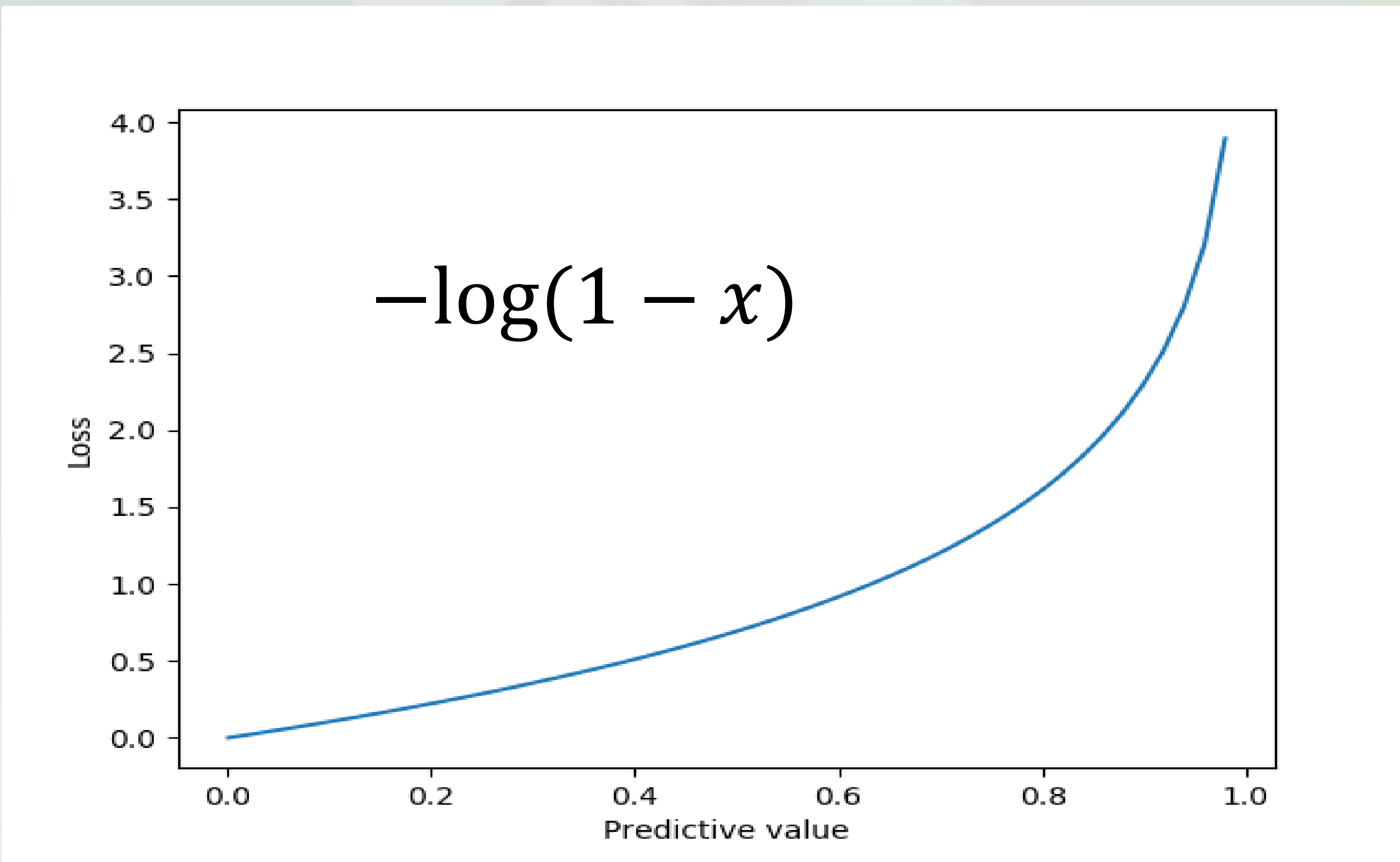
Loss function contruction

Make it nicer



Loss function constrcution

- When $\boldsymbol{y} = \mathbf{0}$, lower $p_{\mathbf{c}}(\mathbf{x})$ is more accurate ⇒ smaller lost
- Any function reflects that behavior?



Loss function construction

- So we have

$$L(p_{\mathbf{c}}(\mathbf{x}), y) = \begin{cases} -\log\left(p_{\mathbf{c}}(\mathbf{x})\right) & \text{if } y = 1 \\ -\log\left(1 - p_{\mathbf{c}}(\mathbf{x})\right) & \text{if } y = 0 \end{cases}$$

- Combine two goals into one function:

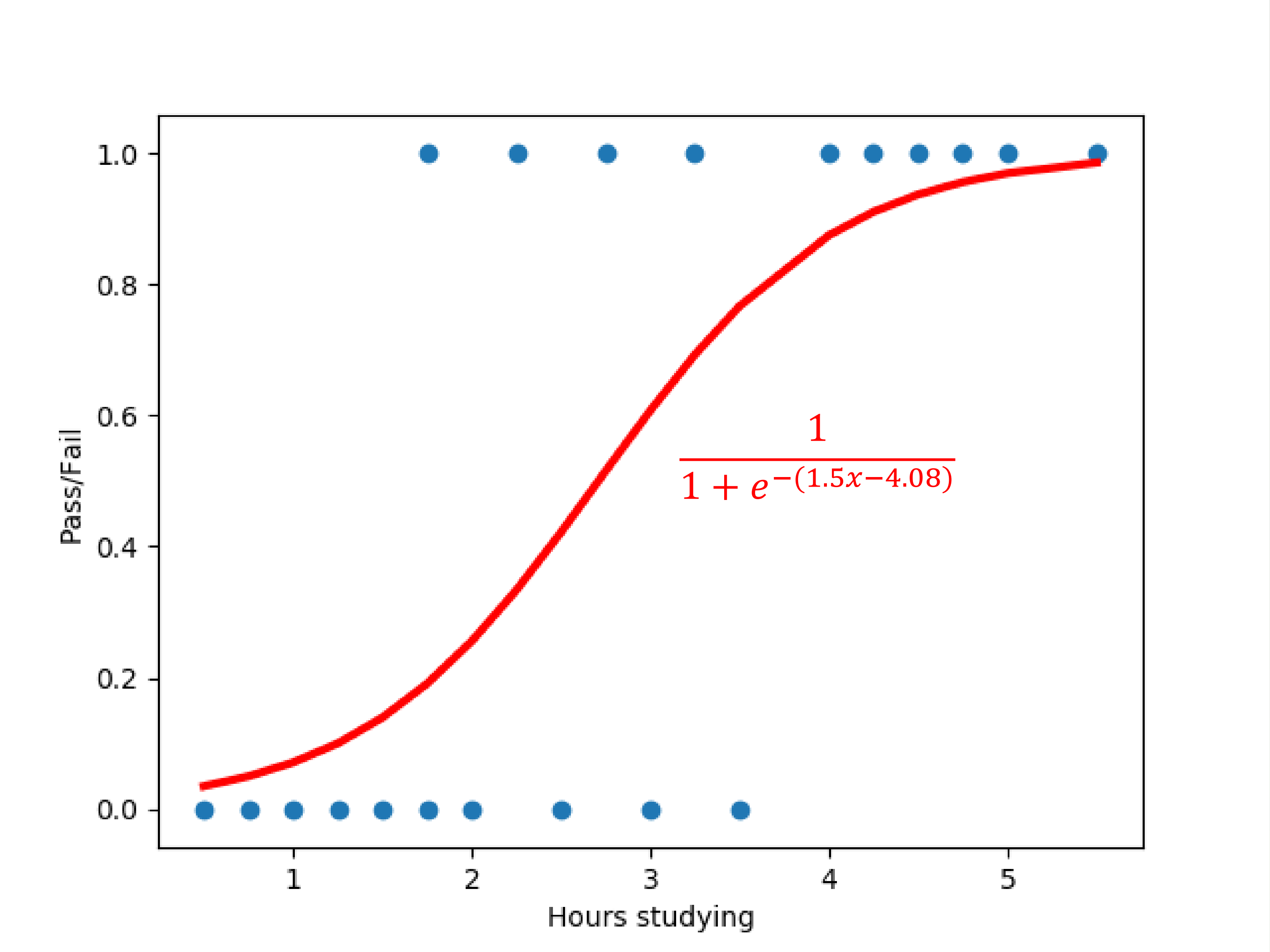
$$L(p_{\mathbf{c}}(\mathbf{x}), y)$$
$$= -y\log\left(p_{\mathbf{c}}(\mathbf{x})\right) - (1 - y)\log\left(1 - p_{\mathbf{c}}(\mathbf{x})\right)$$

Loss function

- Total loss for the whole dataset

$$L(\mathbf{c}) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(p_{\mathbf{c}}\left(\mathbf{x}^{(i)}\right)\right)\right.$$

Resulting Predicted Model



Decision Boundary

- Boundary that separates this class from another ones
- We can construct the **decision boundary** from optimal predictor

$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{c}^T\mathbf{x}}}$$

$$\mathbf{c}^T = (-1,1,1)$$

$$\mathbf{x}^T = (1, x_1, x_2)$$

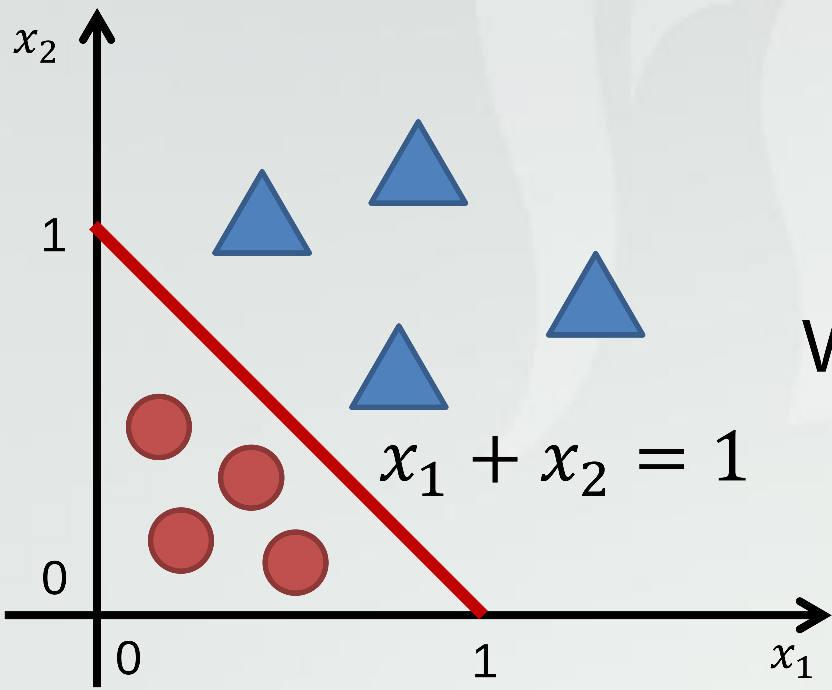
$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1+e^{1-x_1-x_2}}$$

When $p_{\mathbf{c}}(\mathbf{x}) = 0.5$ (threshold)

$$\frac{1}{1+e^{1-x_1-x_2}} = \frac{1}{2}$$

$$\Rightarrow \boxed{x_1 + x_2 = 1}$$

Decision boundary

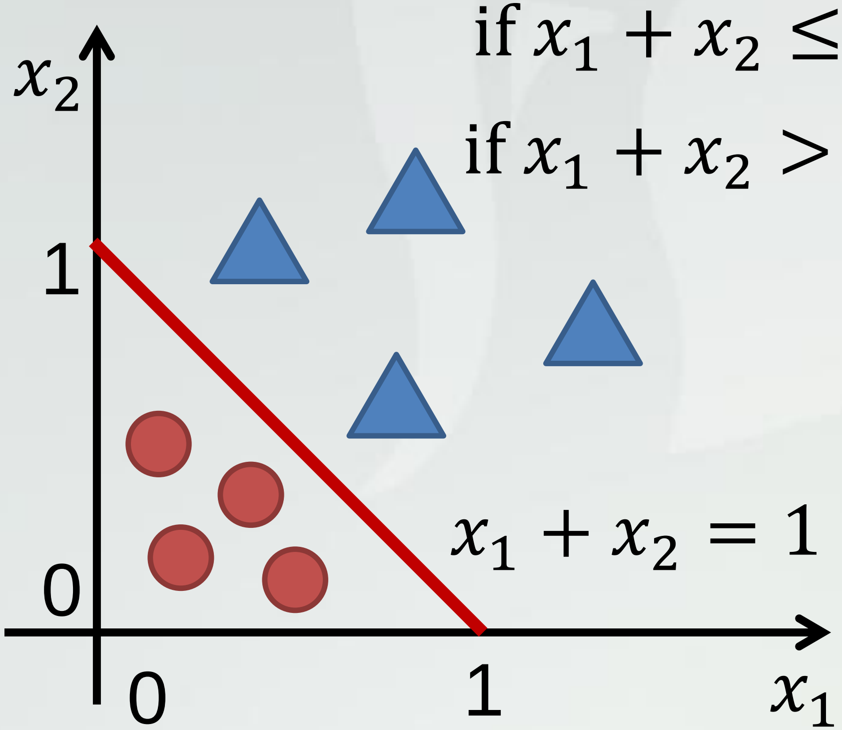


Decision Boundary

- Boundary that separates this class from another ones
- We can construct the decision boundary from optimal predictor

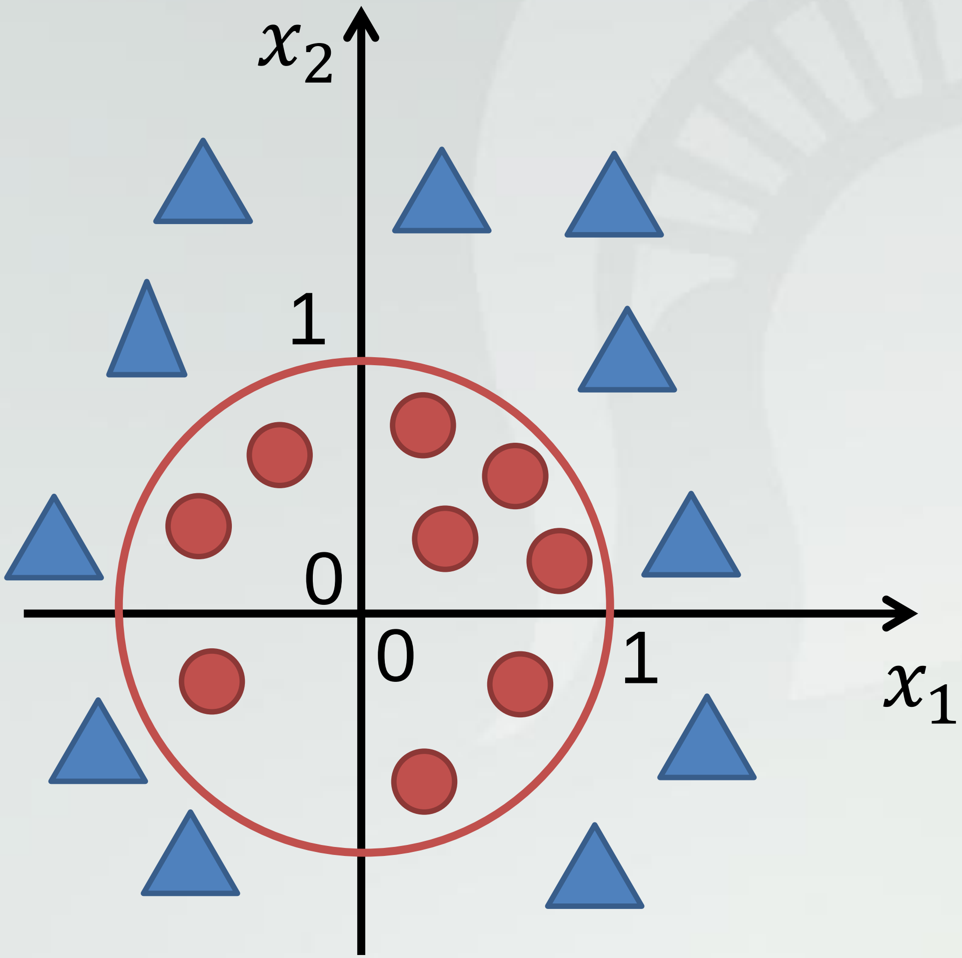
if $x_1 + x_2 \le 1$: data point (x_1, x_2) is

if $x_1 + x_2 > 1$: data point (x_1, x_2) is



Decision Boundary

- Decision boundary can be a curve or manifold



$$p_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1 + e^{1 - x_1^2 - x_2^2}}$$

$$p_{\mathbf{c}}(\mathbf{x}) = 0.5$$

$$\frac{1}{1 + e^{1 - x_1^2 - x_2^2}} = \frac{1}{2}$$

$$\Rightarrow x_1^2 + x_2^2 = 1$$

Multi-Class

- Loss function designs for two classes
- What if we have more than 2 classes?
- Examples:
  - Student Performance: A, B, C, D, F
  - Weather prediction: Sunny, Cloudy, Rain, Snow

Multi-Class

- One-vs-all:
  - If we have *n* classes: $l_1, l_2, \ldots, l_n$
  - For each class l_i. Consider two labels: l_i and not l_i
  - Train logistic regression classifier for this case to get the probability $p_{\mathbf{c}}^{l_i}(\mathbf{x})$ that $y = l_i$
  - Pick the class l_k that maximizes $\max_k p_{\mathbf{c}}^{l_k}(\mathbf{x})$