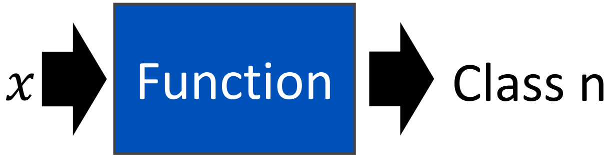
Classification

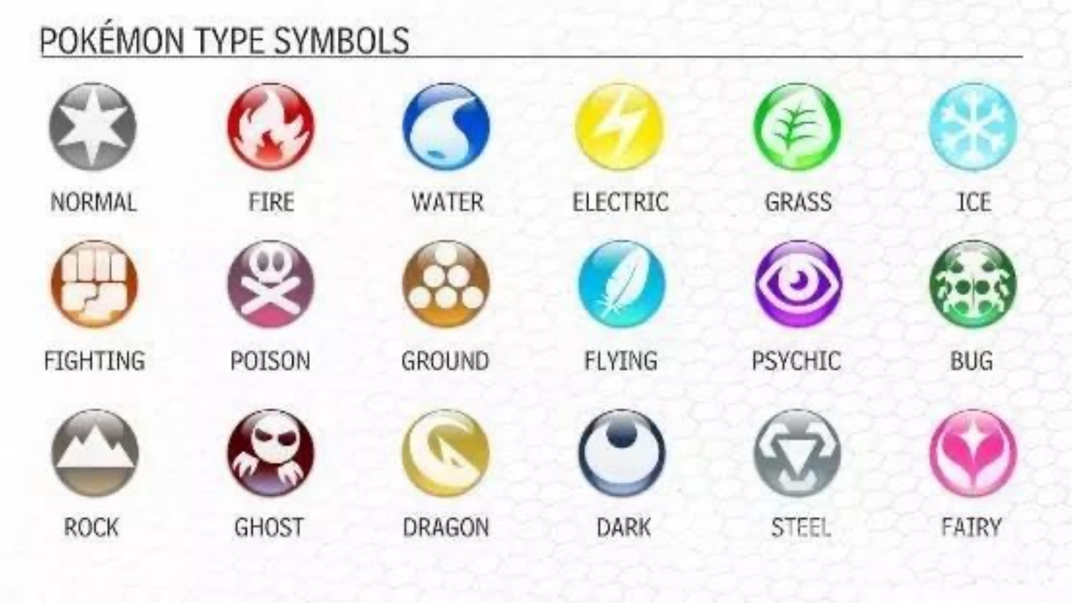
Classification Introduction

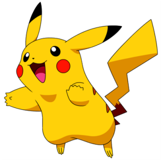
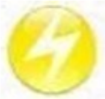
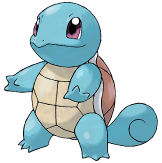
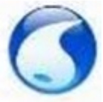
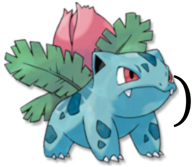
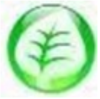


x → Function → Class n

- Credit Scoring
  - Input: income, savings, profession, age and past financial history
  - Output: accept or refuse
- Medical Diagnosis
  - Input: current symptoms, age, gender and past medical history
  - Output: type of diseases
- Handwritten Character Recognition
- Face Recognition
  - Input: image of a face
  - Output: person

Example Application



$f($  $) =$  $f($  $) =$  $f($  $) =$ 

Example Application, Continued

Pokémon games (***not*** Pokémon cards or Pokémon Go)

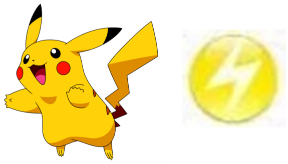
HP: hit points or health. Defines how much damage a Pokémon can withstand before fainting	35
Attack: the base modifier for normal attacks (e.g., Scratch, Punch)	55
Defense: the base damage resistance against normal attacks	40
SP Attack: the base modifier for special attacks (e.g., fire blast, bubble beam)	50
SP Def: the base damage resistance against special attacks	50
Speed: determines which Pokémon attacks first during each round	90

Can we predict the "type" of Pokémon based on the information?

How to do Classification

Training Data for Classification

$(x^1, \hat{y}^1)$



$(x^2, \hat{y}^2)$

……

$(x^N, \hat{y}^N)$

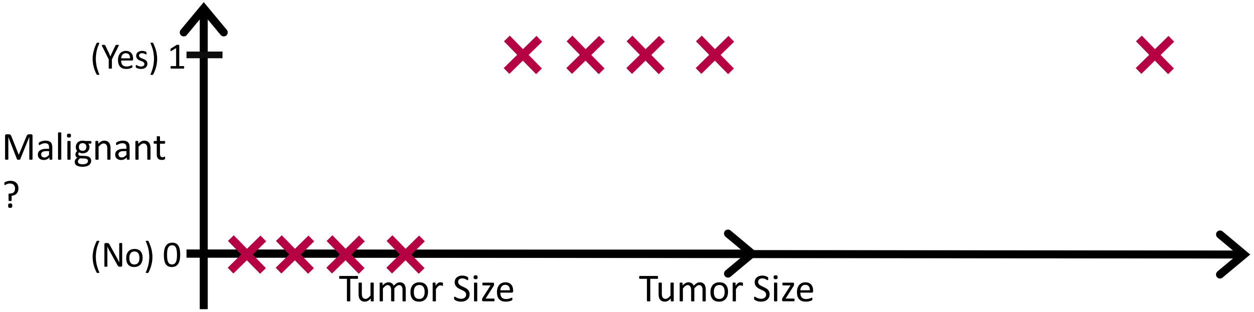
Classification as Regression?

Binary classification as example

Training: Class 1 means the target is 1; Class 2 means the target is -1

Testing: closer to 1 → class 1; closer to -1 → class 2

Using Regression for Classification



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"

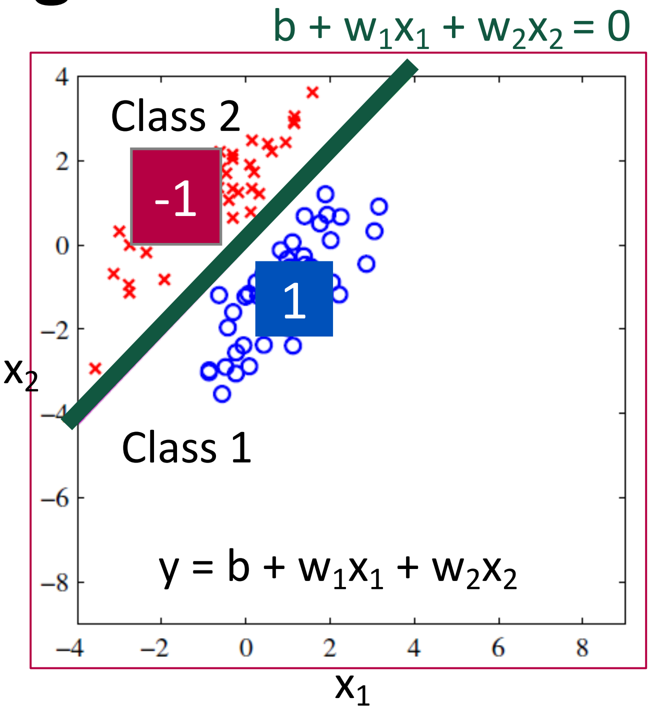
Using Regression for Classification? Continued

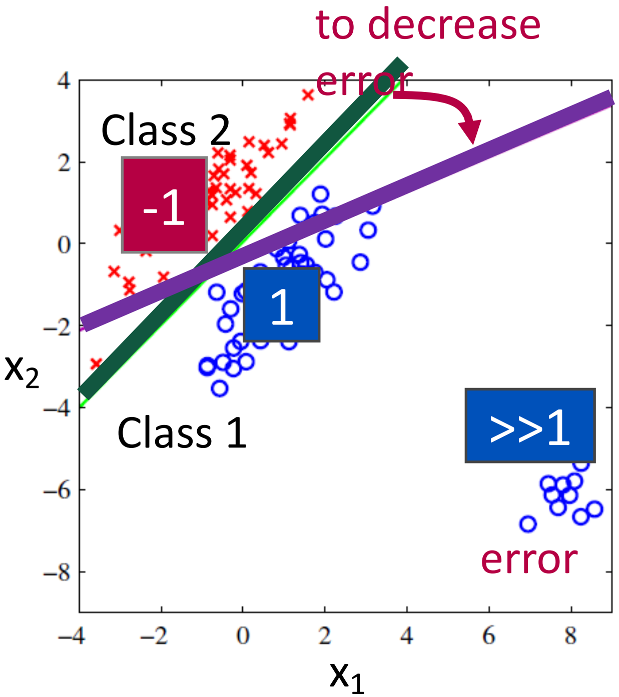
Multiple Classes:

Class 1 means the target is 1

Class 2 means the target is 2

Class 3 means the target is 3 —problematic



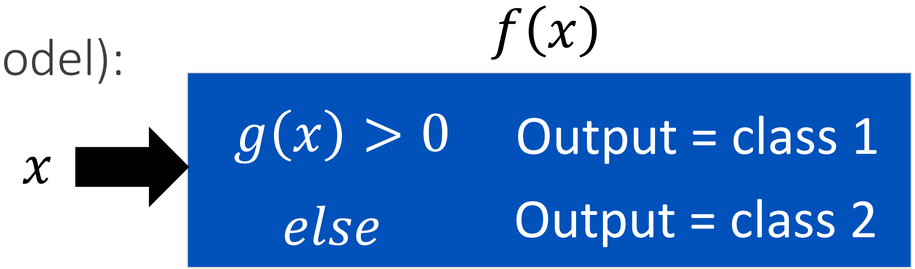


Penalize to the examples that are “too correct”.

(Bishop, 2006, p 186)

Ideal Alternatives

- Function (Model):



$f(x)$

x

$g(x) > 0$ Output = class 1

$else$ Output = class 2

- Loss function:

$$L(f) = \sum_n \delta(f(x^n) \neq \hat{y}^n)$$

The number of times f get incorrect results on training data.

- Find the best function:

Gradient Descent