Machine Learning 1

Logistic Regression

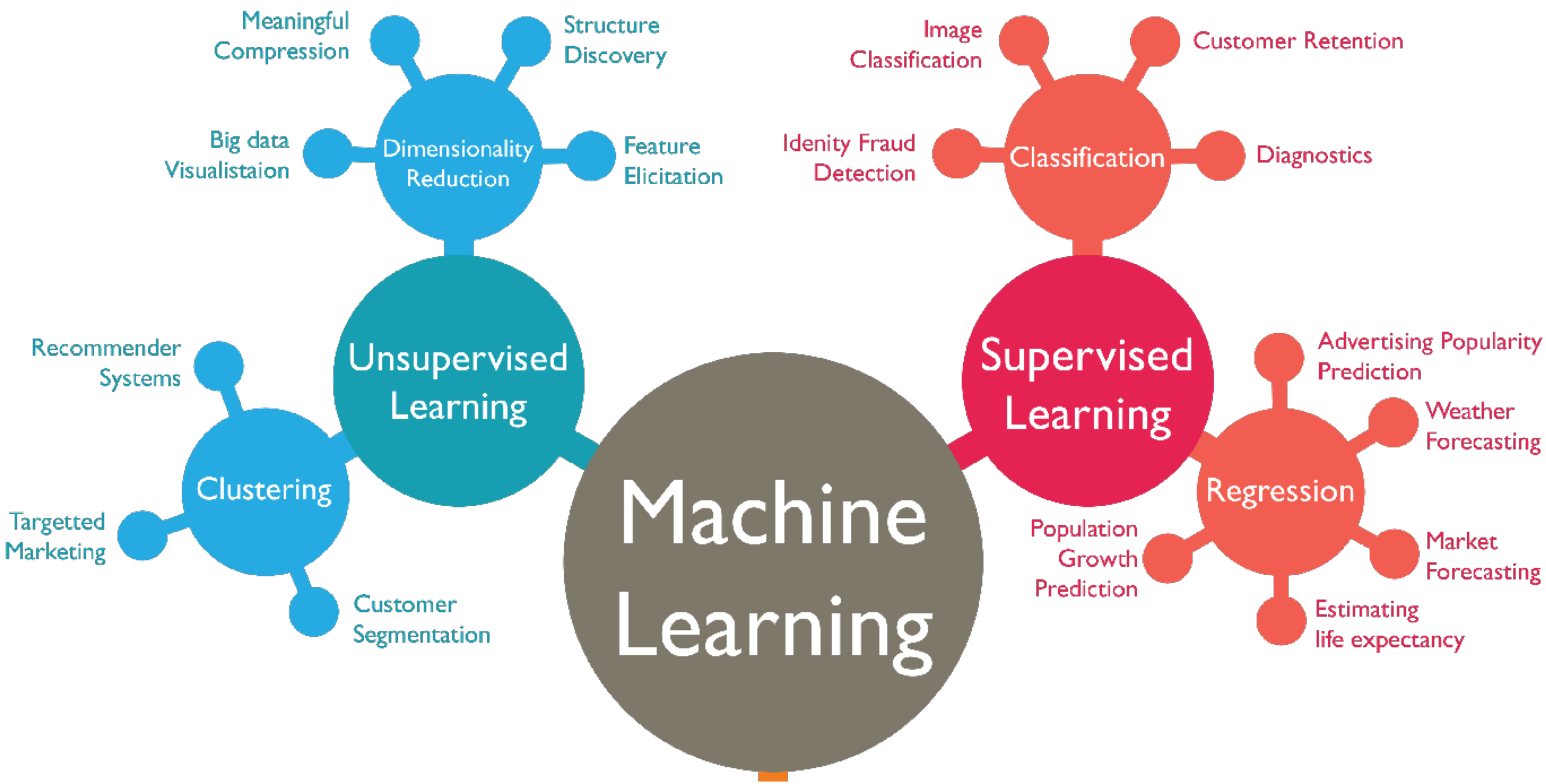
Giovanni Chierchia

Context

- What is **machine learning** ?
  - *The ability of computers to learn without being explicitly programmed*

- There are several types of learning
  - ***Supervised*** ➜ *Teach the computer how to do something*
  - ***Unsupervised*** ➜ *Let the computer learn how to do something*
  - ***Reinforcement*** ➜ *Allow the computer automate decision-making*

A glimpse of machine learning



Supervised learning

Fundamental hypothesis
Generalization by inductive bias
Training process
Regression vs. Classification

Supervised learning

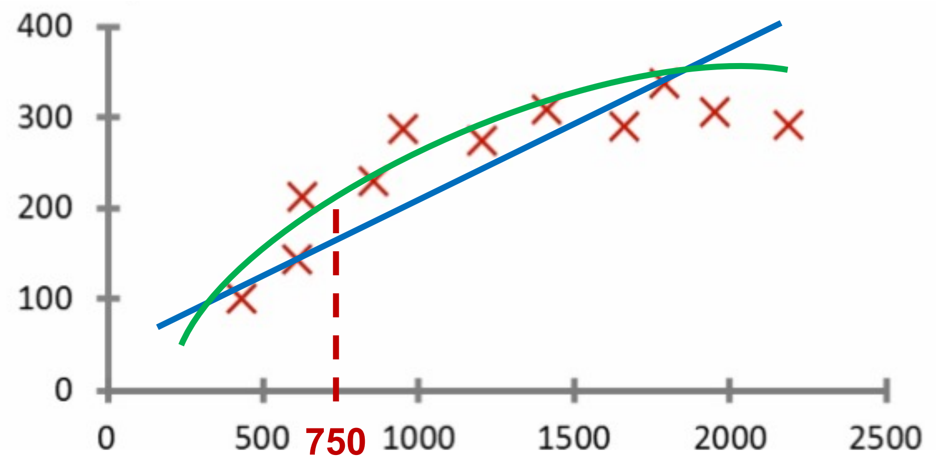
- **Fundamental hypothesis**
  - *Our goal is to predict an output from an input*
  - *We are given a dataset of input-output examples*
  - *We know there is a relationship between the input and the output*

	Input feature 1	*Input feature 2*	*Input feature 3*	*Input feature 4*	*Output*
	Size (feet^2)	**Number of bedrooms**	**Number of floors**	**Age of home (years)**	**Price ($1000)**
$(x^{(1)},y^{(1)})$ = *example 1*	$x_1^{(1)} = 2104$	$x_2^{(1)} = 5$	$x_3^{(1)} = 1$	$x_4^{(1)} = 45$	$y^{(1)} = 460$
$(x^{(2)},y^{(2)})$ = *example 2*	$x_1^{(2)} = 1416$	$x_2^{(2)} = 3$	$x_3^{(2)} = 2$	$x_4^{(2)} = 40$	$y^{(2)} = 232$
$(x^{(3)},y^{(3)})$ = *example 3*	$x_1^{(3)} = 1534$	$x_2^{(3)} = 3$	$x_3^{(3)} = 2$	$x_4^{(3)} = 30$	$y^{(3)} = 315$
$(x^{(4)},y^{(4)})$ = *example 4*	$x_1^{(4)} = 852$	$x_2^{(4)} = 2$	$x_3^{(4)} = 1$	$x_4^{(4)} = 36$	$y^{(4)} = 178$
	…	…	…	…	…

Example

- A friend has a house of 750 square feet
  - *Given the data, how much can he be expected to get?*

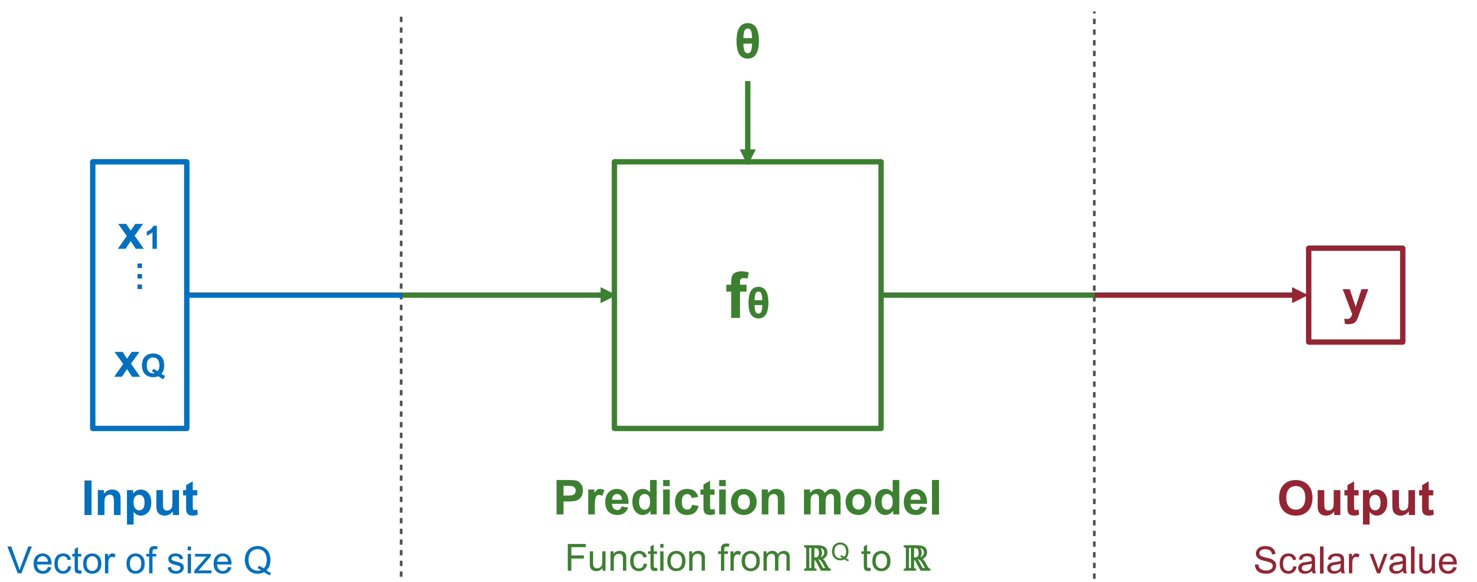
- A learning algorithm can
  - *Fit a straight line through data*
    - *the answer is $150'000*
  - *Fit a second order polynomial*
    - *the answer is $200'000*



- Each of these ones is a supervised learning model !
  - *Later in this course* → *How to chose the best model?*

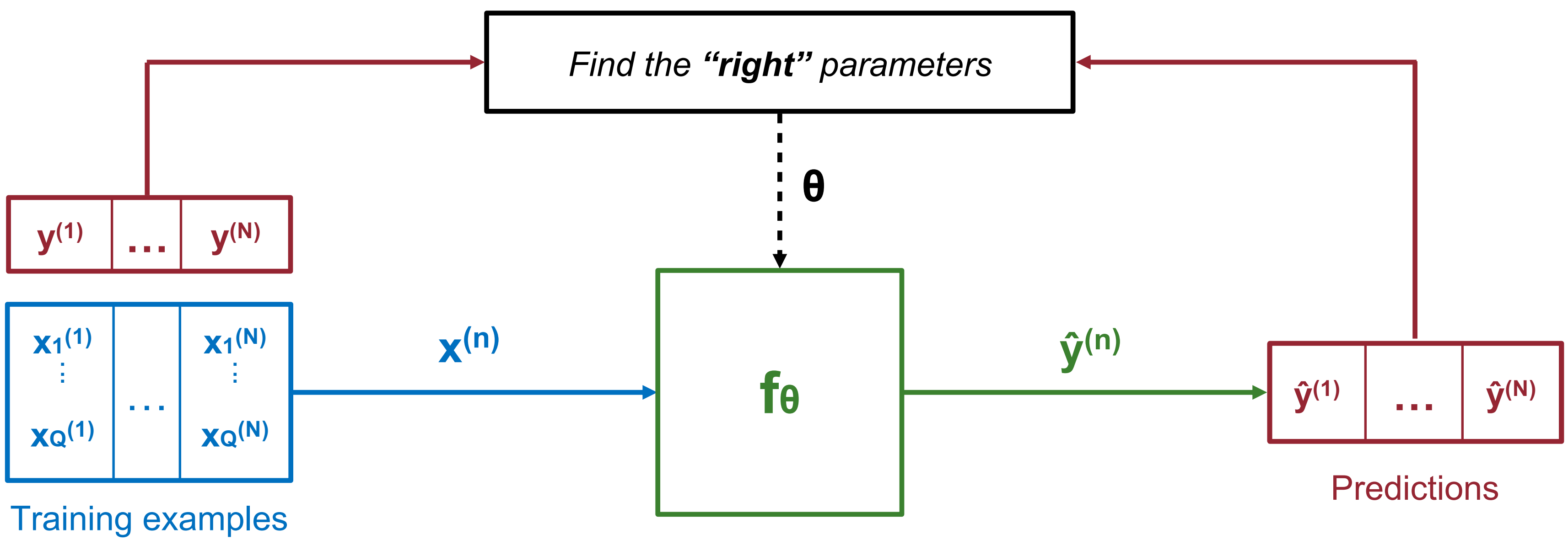
Prediction model

- **Generalization by inductive bias**
  - *We are interested in predicting the output for new unseen inputs*
  - *To do so, we use a parametric model* f_θ *(where* θ *is a vector of parameters)*



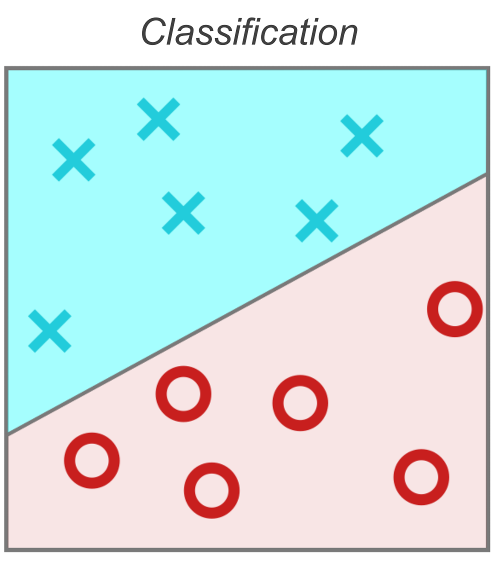
Training process

- **Learning**
  - *Our goal is to learn the prediction model* $\boldsymbol{f_\theta}$ *from training data*
  - *This amounts to finding the "right values" for parameters* $\boldsymbol{\theta}$

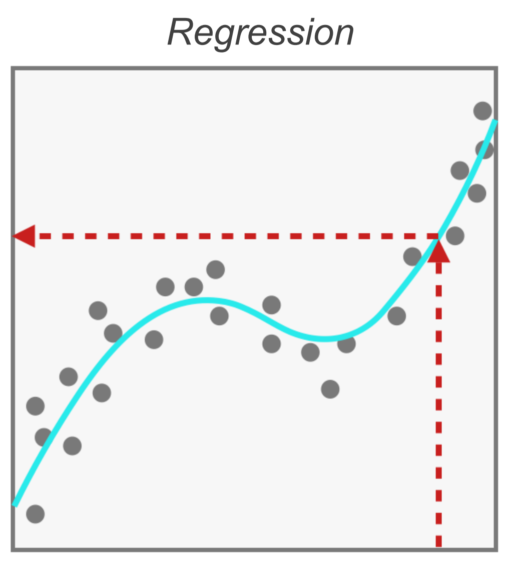


Supervised learning

- **Two types of problems**
  - ***Regression*** ➔ *Learning how to predict a* ***continuous*** *output*
  - ***Classification*** ➔ *Learning how to predict a* ***discrete*** *output*



Here, the line classifies the observations into X's and O's



Here, the fitted line provides a predicted output, if we give it an input

Classification

Quantitative response

Qualitative response

Why not linear regression ?

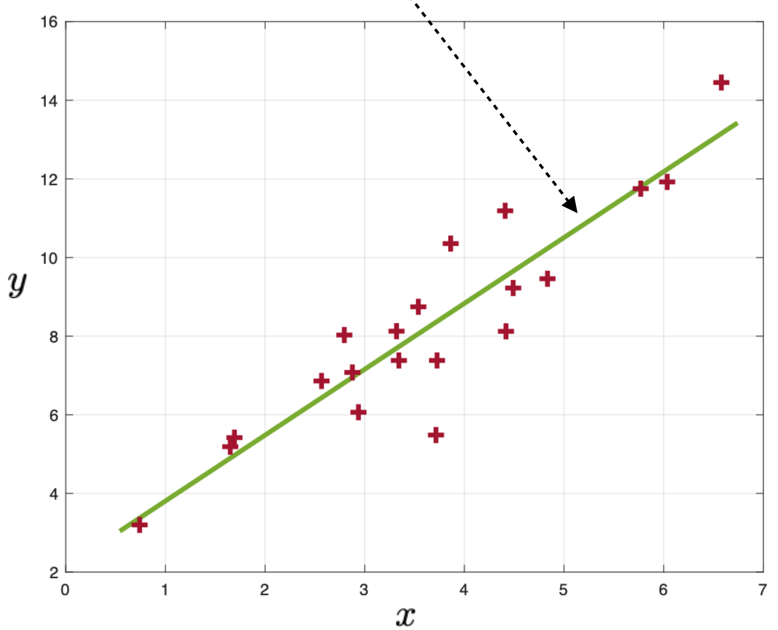
Quantitative response (1/2)

- Suppose we are provided with some **advertising data**
  - *Product sales **AND** their advertising budgets for TV, radio, …*
- Based of this data, we are asked to suggest a marketing plan for next year that will result in high product sales.
  - *Is there a relationship between advertising budgets and sales?*
  - *How accurately can we predict future sales?*
  - *Is there synergy among the advertising media?*
- **Linear regression** can be used to answer these questions

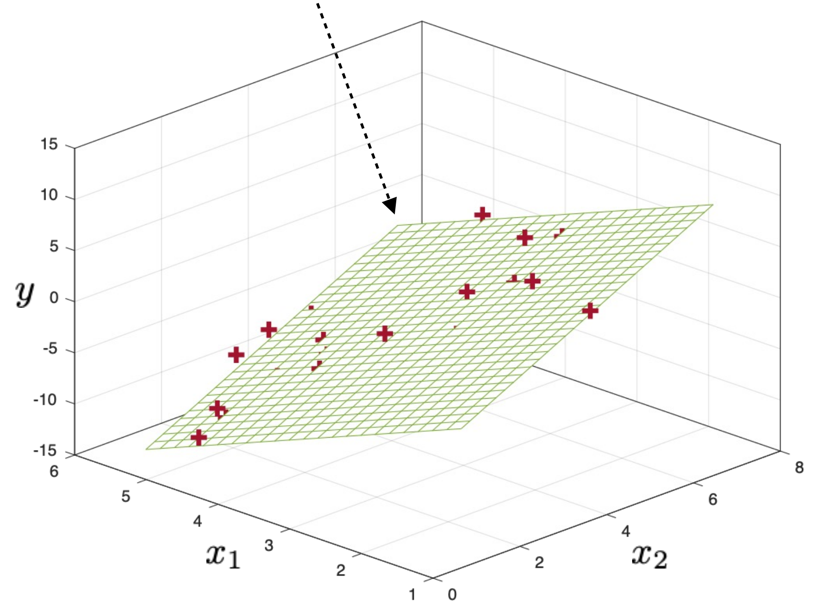
Quantitative response (2/2)

- Linear regression can make a **quantitative prediction**

$$f_\theta(x) = \theta_0 + \theta_1 x$$



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



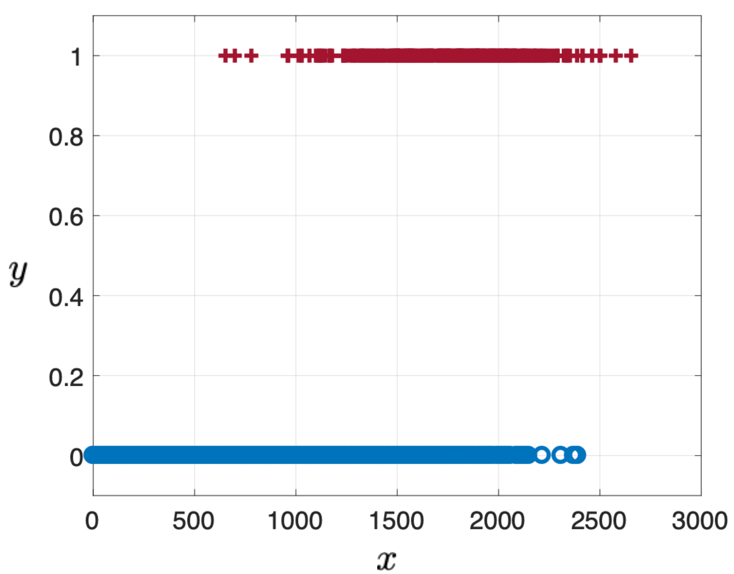
Qualitative response (1/2)

- But in many situations, the prediction must be **qualitative**
  - *A person suffers from symptoms that could possibly be attributed to one of three medical conditions. Which of them does he have?*
  - *An online banking service needs to determine whether or not a transaction being performed on the site is fraudulent, on the basis of the user's IP address, past transaction history, and so forth.*
  - *On the basis of DNA sequence data for a number of patients with and without a given disease, a biologist would like to figure out which DNA mutations are causing diseases and which are not.*
- Linear regression **can't be used** to answer these questions

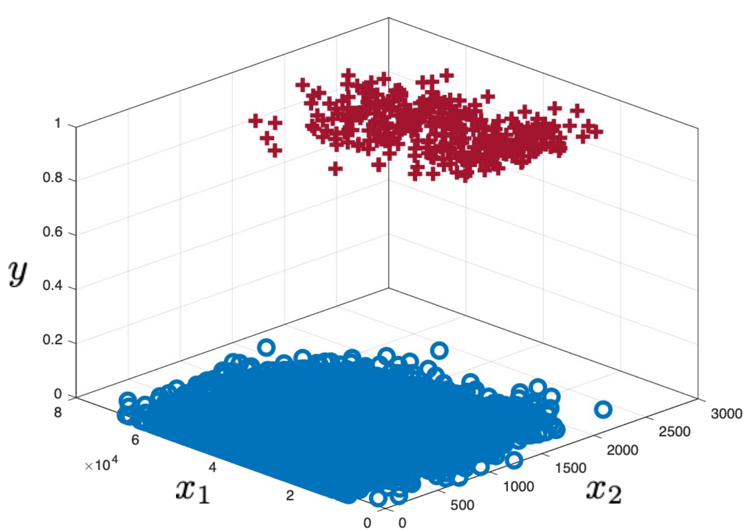
Qualitative response (2/2)

- A **qualitative prediction** is equivalent to **classification**
  - *It is the task of assigning an observation to a category, or class.*

Binary classification (1D)



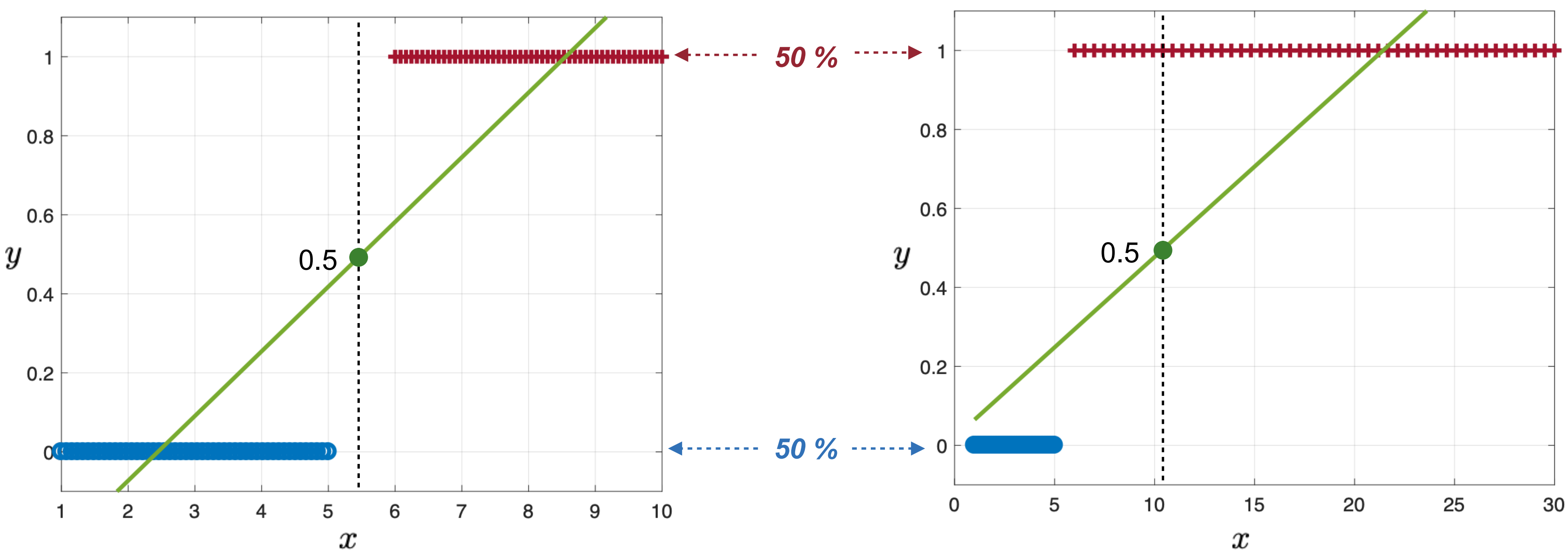
Binary classification (2D)



Why not linear regression ? (1/2)

- Linear regression **performs poorly** in classification
  - *Learning is sensitive to distribution of data <u>within</u> classes*

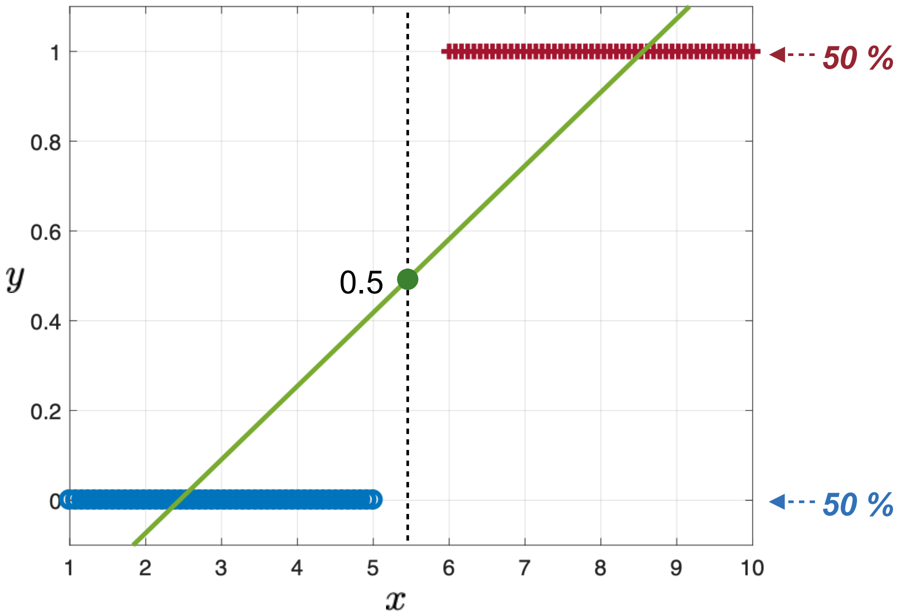
Similar distributions | *Different distributions*



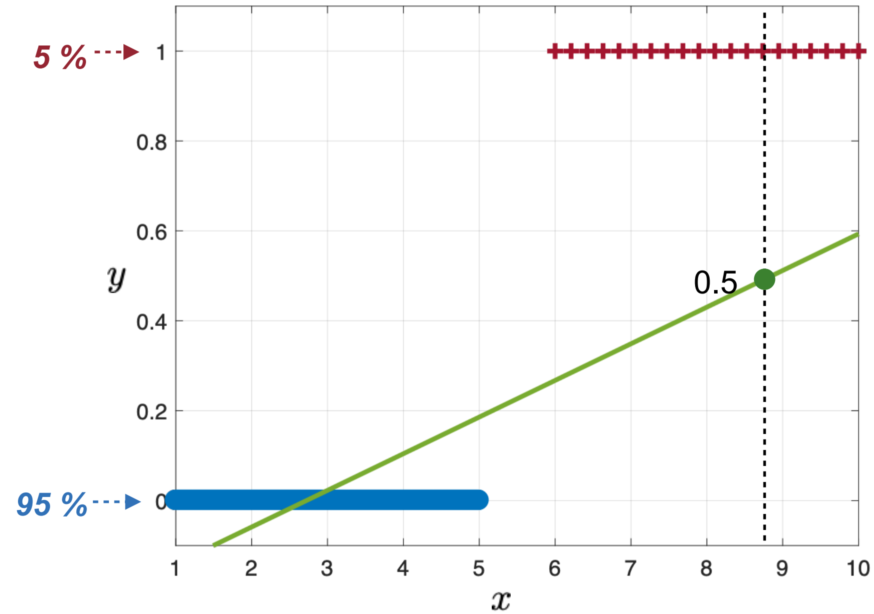
Why not linear regression ? (2/2)

- Linear regression **performs poorly** in classification
  - *Learning is sensitive to distribution of data between classes*

Balanced classes



Unbalanced classes



Quiz

- You are running a company, and you want to develop learning algorithms to address the following problems.
  1. *You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 monts.*
  2. *You need to examine individual customer accounts, and for each one decide if it has been hacked/compromised.*

Should you treat them as classification or regression ?

A. Treat both problems as classification

B. Treat problem 1 as classification and problem 2 as regression

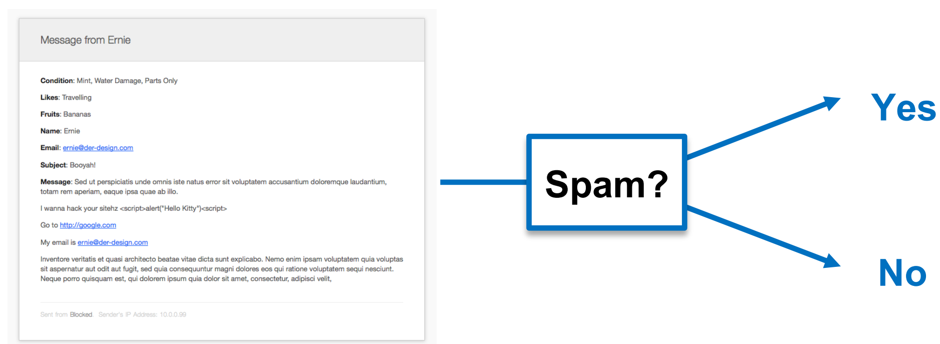
C. Treat problem 1 as regression and problem 2 as classification

D. Treat both problems as regression

What we have seen so far…

- Supervised learning can be categorized into
  - *regression* → *learning how to predict a **continuous** response*
  - *classification* → *learning how to predict a **discrete** response*

EXAMPLE:



- Linear regression isn't good for making discrete predictions
  - ***In this lecture*** → *How to deal with classification ?*

Logistic model

Binary classification

Logistic function

Interpretation

Binary classification (1/2)

- For now, we focus on classification with **two classes**
  - *the response variable **y** is a binary value*

positive class

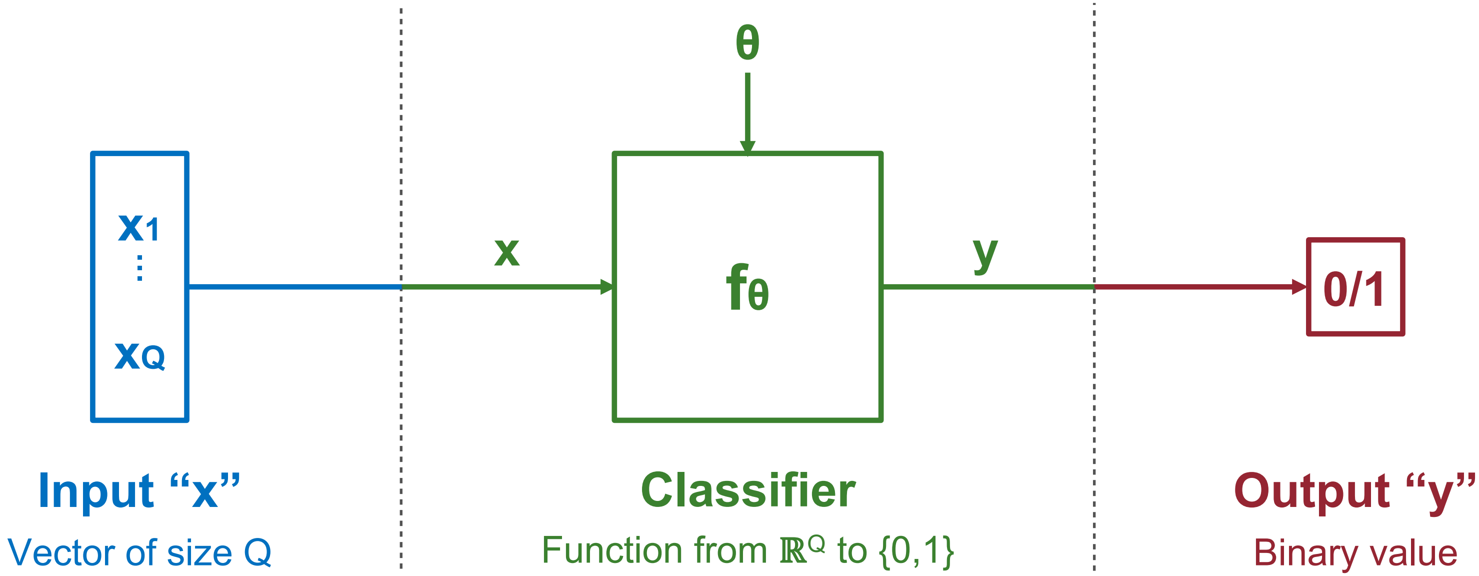
$$y \in \{0, 1\}$$

negative class

- Examples
  - *email → spam / not spam ?*
  - *online transaction → fraudulent (yes / no) ?*
  - *tumor → malignant / benign ?*

Binary classification (2/2)

- Our goal is to **predict** the class **y** from an observation **x**
  - *To do so, we use a parametric model* $\boldsymbol{f_\theta}$ *…*
  - *… where* $\boldsymbol{\theta = [\theta_0, \theta_1, \ldots, \theta_Q]^T}$ *is a vector of parameters to be estimated.*



Logistic function (1/3)

- How to **predict** a binary response variable ?
  - *Actually, we don't directly predict a binary outcome*
  - *Instead, we predict the **probability** that **y = 1** given **x***

$$f_\theta(\mathrm{x}) \approx \mathsf{P}(y = 1 \,|\, \mathrm{x})$$

  - *To do so, we use a **bounded** linear model*

$$f_\theta(\mathrm{x}) = g\big(\theta_0 + \theta_1 x_1 + \cdots + \theta_Q x_Q\big)$$

  - *where **g** is the **logistic function***

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

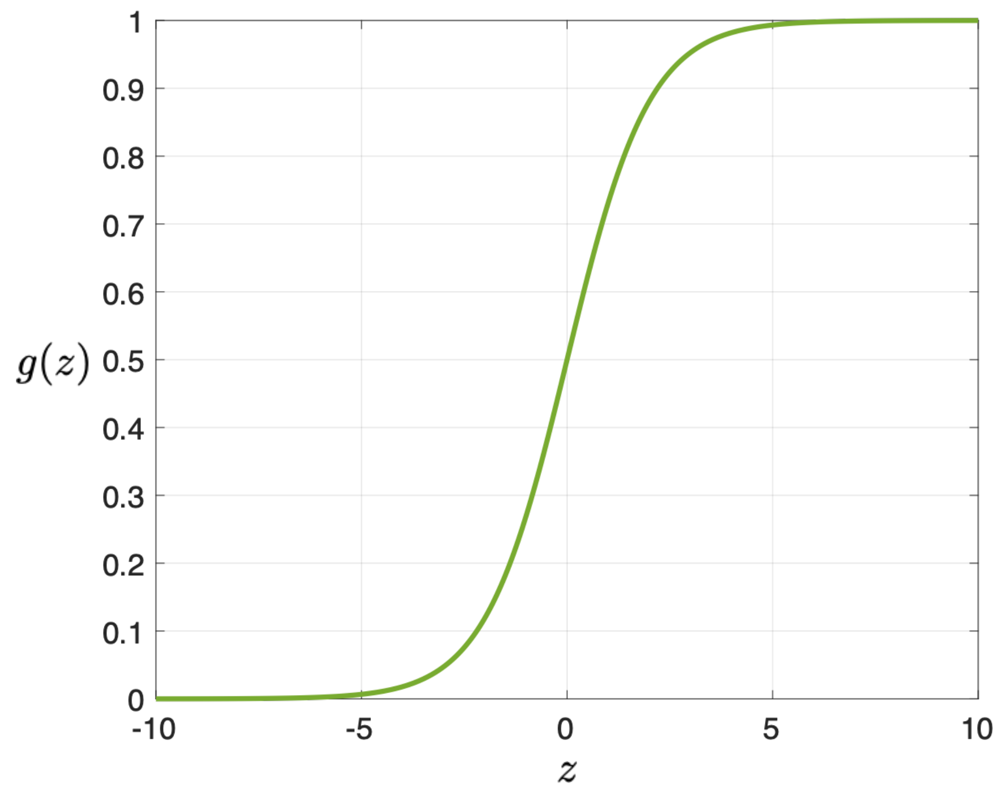
Logistic function (2/3)

- The **logistic function** maps a real value between **0** and **1**
  - *Hence, it can be regarded as a probability.*

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

Properties

$$g(z) = \frac{e^z}{1+e^z}$$
$$g(-z) = 1 - g(z)$$
$$g'(z) = g(z)\big(1-g(z)\big)$$
$$g^{-1}(t) = \log\left(\frac{t}{1-t}\right)$$



Logistic function (3/3)

- Logistic model will be compactly written as

$$f_\theta(\mathrm{x}) = \frac{1}{1+\exp(-\theta^\top \mathrm{x})}$$

  - *NOTE 1:* ***x*** *and* ***θ*** *are column vectors of size Q+1 (with* $\mathbf{x_0}$ ***= 1****)*

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_Q \end{bmatrix} \qquad \mathrm{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_Q \end{bmatrix}$$

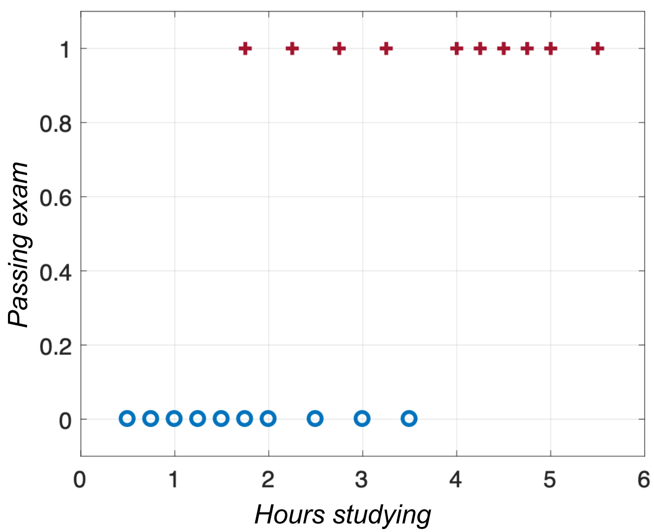
  - *NOTE 2: the linear combination of* ***x*** *and* ***θ*** *is a scalar product*

$$\theta^\top \mathrm{x} = [\theta_0 \; \theta_1 \; \dots \; \theta_Q] \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_Q \end{bmatrix} = \theta_0 + \theta_1 x_1 + \cdots + \theta_Q x_Q$$

Prediction (1/2)

- Suppose we wish to answer the following question.
  - *A group of 20 students studied between 0 and 6 hours for an exam.*
  - *How does the number of hours spent studying affect the probability that the student will pass the exam?*

Training data



Learning →

Logistic model

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

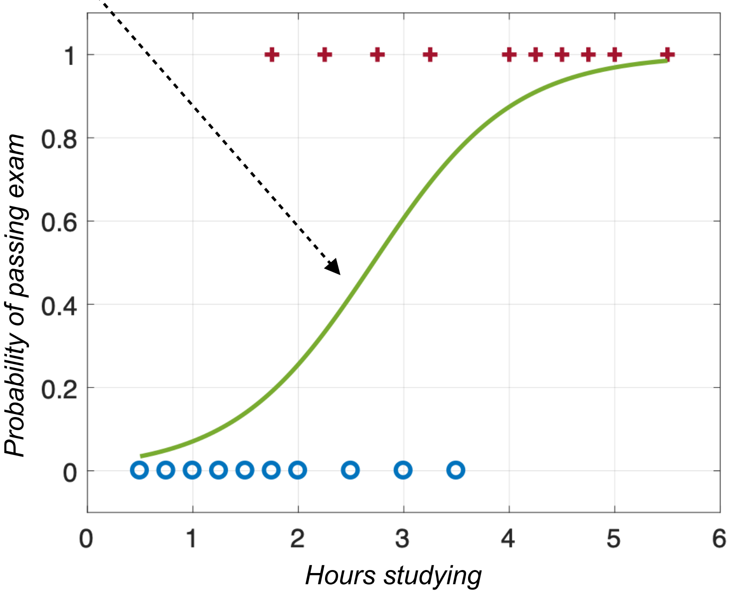
Prediction (2/2)

- Learning yields the following parameters
  - *We will see later how to do this*

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} -4.07 \\ 1.50 \end{bmatrix}$$

$$\text{Prob. of passing exam} = \frac{1}{1 + \exp(4.07 - 1.50 \times \text{Hours studying})}$$

Hours studying	Prob. of passing exam
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97
6	Compute it yourself !



Odds (1/3)

- Logistic model can be related to the **odds for y=1**

$$\mathsf{odds}(\mathrm{x}) = \frac{\mathsf{P}(y=1\,|\,\mathrm{x})}{\mathsf{P}(y=0\,|\,\mathrm{x})} = \frac{f_\theta(\mathrm{x})}{1-f_\theta(\mathrm{x})} = e^{\theta_0+\theta_1 x_1+\cdots+\theta_Q x_Q}$$

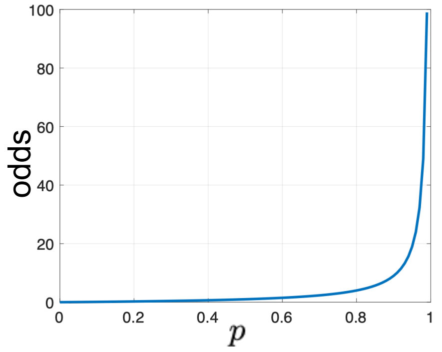
- The odds ratio provides an interpretation for parameter $\boldsymbol{\theta_i}$
  - *one unit increase in variable* $\boldsymbol{x_i}$ *multiplies the odds by* $\boldsymbol{\exp(\theta_i)}$

$$\frac{\mathrm{odds}(x_1,\ldots,x_i+1,\ldots,x_Q)}{\mathrm{odds}(x_1,\ldots,\quad x_i\quad,\ldots,x_Q)} = e^{\theta_i}$$

Odds (2/3) **[Optional]**

- The **odds** are the probability ratio of an event
  - *it reflects the likelihood that the event will take place*

$$\text{odds} = \frac{\mathsf{P}(y=1)}{\mathsf{P}(y=0)} = \frac{p}{1-p}$$



  - *What are the odds for an event whose probability is **5/7**?*

Odds = (5/7) / (1 - 5/7) = 5 / 2

  - *What are the odds for picking a face card (J, Q, K) from a deck of cards?*

Odds = 12 / 40 = 3 / 10

  - *What are the odds for rolling a number greater than 3 on a fair die?*

Odds = 3 / 3 = 1

Odds (3/3) **[Optional]**

- The odds are widely used in gambling
  - *They represent the payout on the stake*

$$\text{winnings} = \left(1 + \frac{1}{\text{odds}}\right) \times \text{stake} = \frac{1}{\mathsf{P}(y=1)} \times \text{stake}$$

  - *An event whose probability is 5/7 ?*

Winnings = 1.4 × stake

  - *Picking a face card (J, Q, K) at random from a deck of cards ?*

Winnings = 4.3 × stake

  - *Rolling a number greater than 3 on a die ?*

Winnings = 2 × stake

Model justification (1/2) **[Optional]**

- Why does the logistic model **predict a probability** ?
  - *Let's apply Bayes' theorem…*

$$\mathsf{P}(y=1\,|\,\mathrm{x}) = \frac{\mathsf{P}(y=1)\,\mathsf{P}(\mathrm{x}\,|\,y=1)}{\mathsf{P}(y=1)\,\mathsf{P}(\mathrm{x}\,|\,y=1)+\mathsf{P}(y=0)\,\mathsf{P}(\mathrm{x}\,|\,y=0)}$$

  - *… and manipulate its expression*

$$\mathsf{P}(y=1\,|\,\mathrm{x}) = \frac{1}{1+\frac{\mathsf{P}(y=0)}{\mathsf{P}(y=1)}\frac{\mathsf{P}(\mathrm{x}\,|\,y=0)}{\mathsf{P}(\mathrm{x}\,|\,y=1)}}$$

  - *HYPOTHESIS 1. Input features are statistical independents*

$$\mathsf{P}(y=1\,|\,\mathrm{x}) = \frac{1}{1+\frac{\mathsf{P}(y=0)}{\mathsf{P}(y=1)}\frac{\mathsf{P}(\mathrm{x}_1\,|\,y=0)}{\mathsf{P}(\mathrm{x}_1\,|\,y=1)}\cdots\frac{\mathsf{P}(\mathrm{x}_Q\,|\,y=0)}{\mathsf{P}(\mathrm{x}_Q\,|\,y=1)}}$$

Model justification (2/2) **[Optional]**

- *HYPOTHESIS 2. Log-likelihood ratio depends linearly on* $\boldsymbol{x_i}$

$$\log\left(\frac{\mathsf{P}(x_i \,|\, y=0)}{\mathsf{P}(x_i \,|\, y=1)}\right) = -\theta_i x_i$$

- *HYPOTHESIS 3. Log-prior ratio is constant*

$$\log\left(\frac{\mathsf{P}(y=0)}{\mathsf{P}(y=1)}\right) = -\theta_0$$

- *Putting all together… we get the logistic model !*

$$\mathsf{P}(y=1 \,|\, \mathrm{x}) = \frac{1}{1 + e^{-\theta_0}\, e^{-\theta_1 x_1} \dots e^{-\theta_Q x_Q}}$$

Quiz

- Suppose you have trained a logistic model, and it outputs on a new example **x** a prediction $\mathbf{f_\theta(x) = 0.7}$. This means (check all that apply):

1) *Our estimate for* ***P(y=1|x)*** *is 0.7.*
2) *Our estimate for* ***P(y=1|x)*** *is 0.3.*
3) *Our estimate for* ***P(y=1|x)*** *is 0.3 × 0.7.*
4) *Our estimate for* ***P(y=0|x)*** *is 0.7.*
5) *Our estimate for* ***P(y=0|x)*** *is 0.3.*
6) *Our estimate for* ***P(y=0|x)*** *is* 0.7^2.
7) *Our estimate for* ***P(y=0|x)*** *is 0.3 × 0.7.*

What we have seen so far…

- Logistic model

$$f_\theta(\mathrm{x}) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \cdots + \theta_Q x_Q)}}$$

- Interpretation of predicted values

$$f_\theta(\mathrm{x}) \approx \mathsf{P}(y = 1 \,|\, \mathrm{x})$$

- Interpretation of model parameters

$$e^{\theta_i} = \frac{\mathrm{odds}(x_1, \dots, x_i + 1, \dots, x_Q)}{\mathrm{odds}(x_1, \dots, \quad x_i \quad, \dots, x_Q)}$$

Logistic regression

Training data

Learning

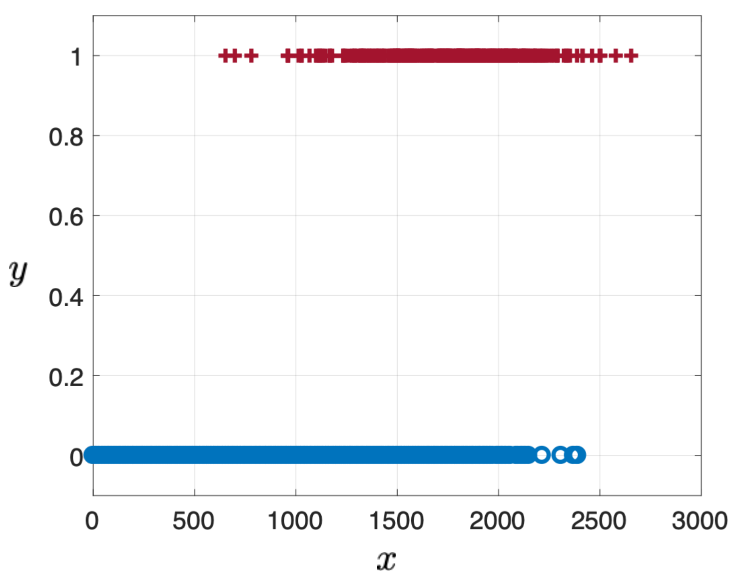
Cost function

Training data (1/2)

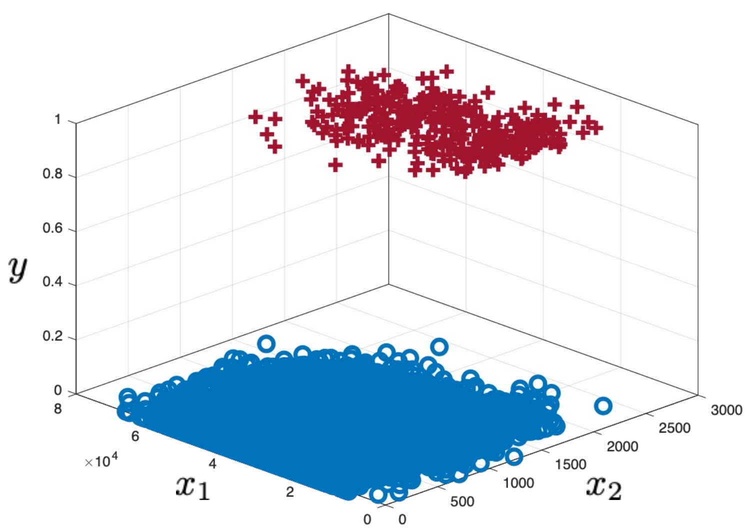
- We are given a set of input-output pairs

$$\left(\mathrm{x}^{(n)}, y^{(n)}\right) \in \mathbb{R}^Q \times \{0,1\} \qquad n = 1,\dots,N$$

Binary classification (Q=1)



Binary classification (Q=2)



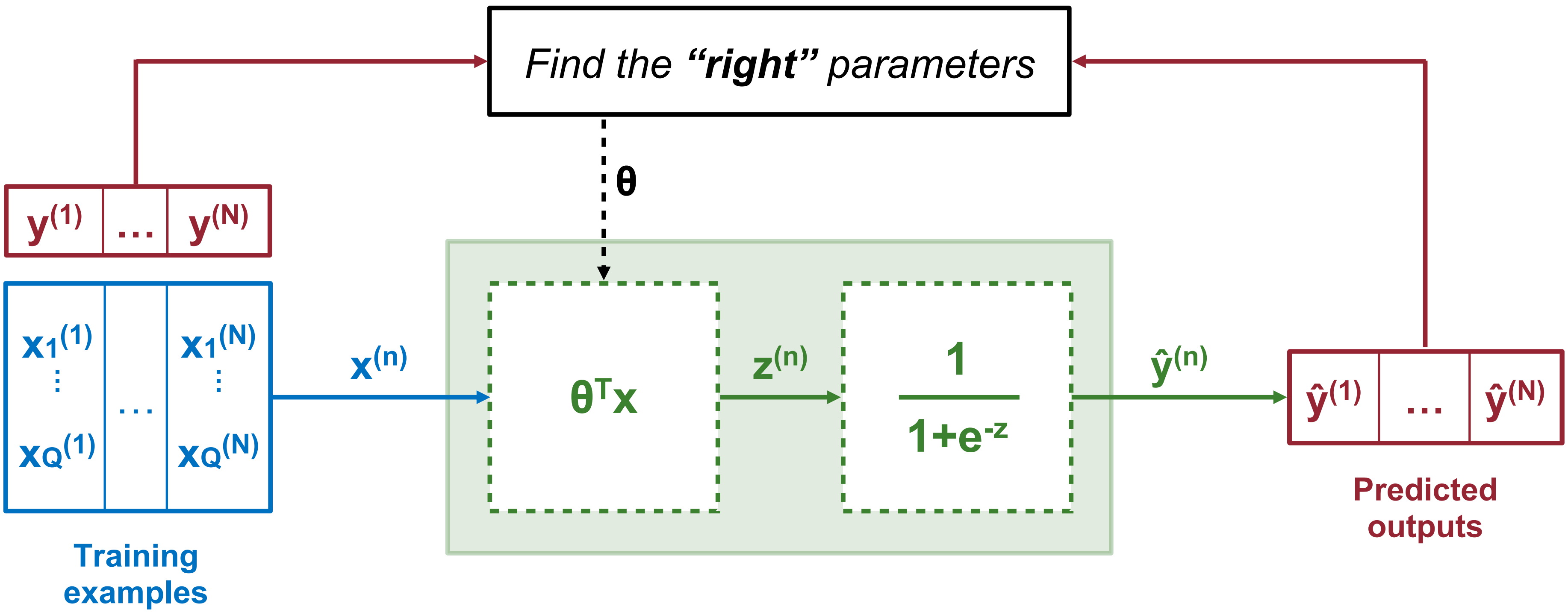
Training data (2/2)

- Notation
  - Q → *number of input features*
  - N → *number of training examples*
  - $x^{(n)}$ → *input vector of the **n**-th training example*
  - $x_i^{(n)}$ → *value of feature **i** in the **n**-th training example*

	Feature 1	*Feature 2*	*Feature 3*	*Output*
	Income	**Student**	**Balance**	**Default**
$(x^{(1)},y^{(1)})$ = *example 1*	$x_1^{(1)} = 44362$	$x_2^{(1)} = 0$	$x_3^{(1)} = 729$	$y^{(1)} = 0$
$(x^{(2)},y^{(2)})$ = *example 2*	$x_1^{(2)} = 12106$	$x_2^{(2)} = 1$	$x_3^{(2)} = 817$	$y^{(2)} = 0$
$(x^{(3)},y^{(3)})$ = *example 3*	$x_1^{(3)} = 17854$	$x_2^{(3)} = 1$	$x_3^{(3)} = 1487$	$y^{(3)} = 1$
$(x^{(4)},y^{(4)})$ = *example 4*	$x_1^{(4)} = 44998$	$x_2^{(4)} = 0$	$x_3^{(4)} = 2033$	$y^{(4)} = 1$
	…	…	…	…

Learning (1/2)

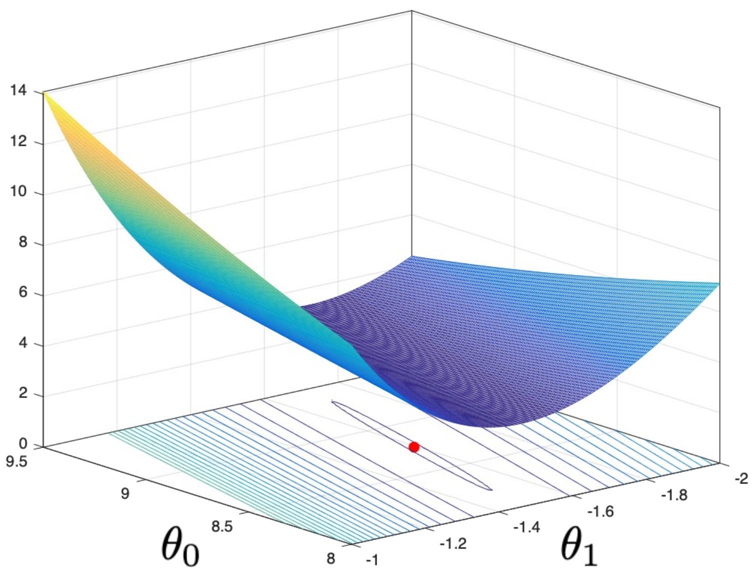
- Our goal is to **learn P(y=1|x)** from training data
  - *This amounts to finding the "right values" of* $\boldsymbol{\theta}$ *in the logistic model*

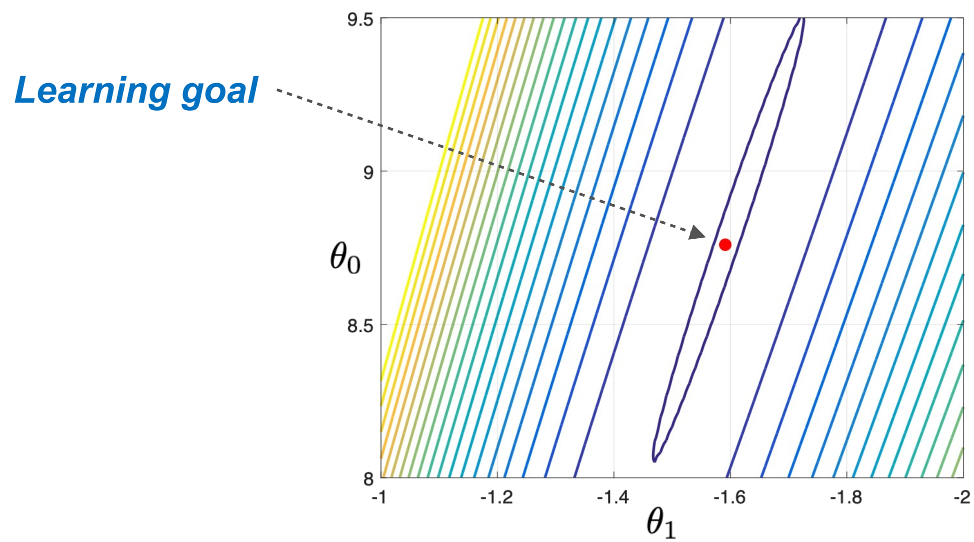


Learning (2/2)

- How to choose the “right values” for parameters **θ** ?
  - *We select **θ** such that the **model f_θ is fitted** to training data*

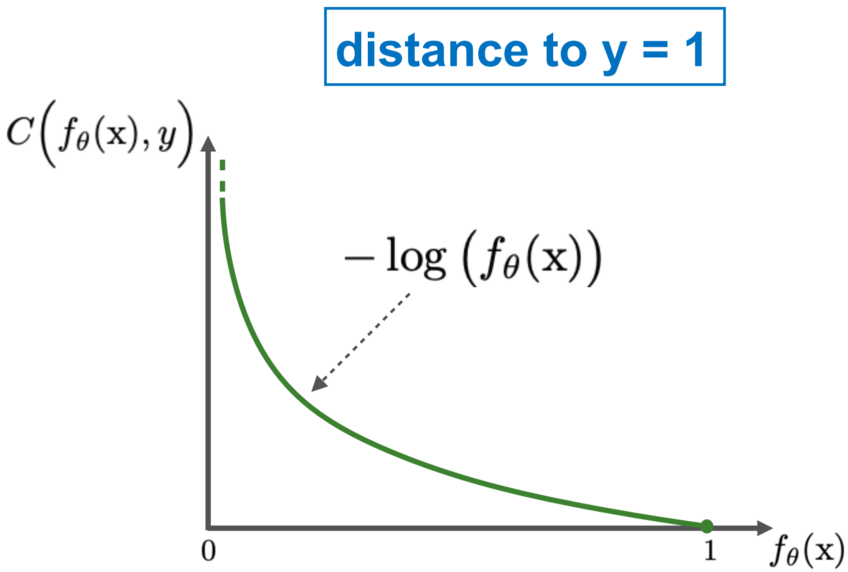
$$\widehat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} C\Big(f_\theta(\mathrm{x}^{(n)}), y^{(n)}\Big)$$

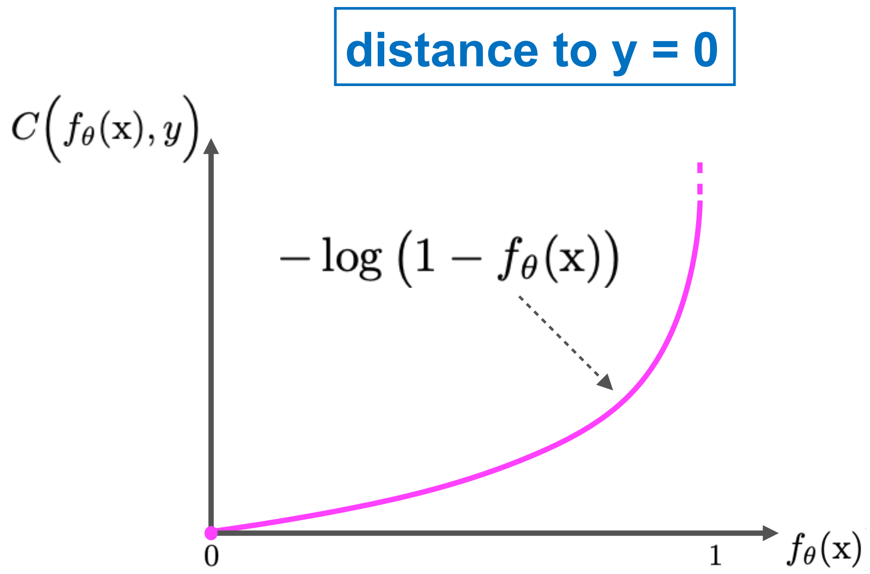




Cost function (1/2)

- How to measure the **fitting of f_θ** to the training data?
  - *for each example* ***(x, y)****, the prediction* ***f_θ(x)*** *must be close to* ***y***
  - *since* ***0 < f_θ(x) < 1****, the distance between* ***f_θ(x)*** *and* ***y*** *can be measured as*





Cost function (2/2)

- Data fitting is quantified by the **logarithm cost function**

$$C\Big(f_\theta(\mathrm{x}), y\Big) = \begin{cases} -\log\big(f_\theta(\mathrm{x})\big) & \text{if } \; y = 1 \\ -\log\big(1 - f_\theta(\mathrm{x})\big) & \text{if } \; y = 0 \end{cases}$$

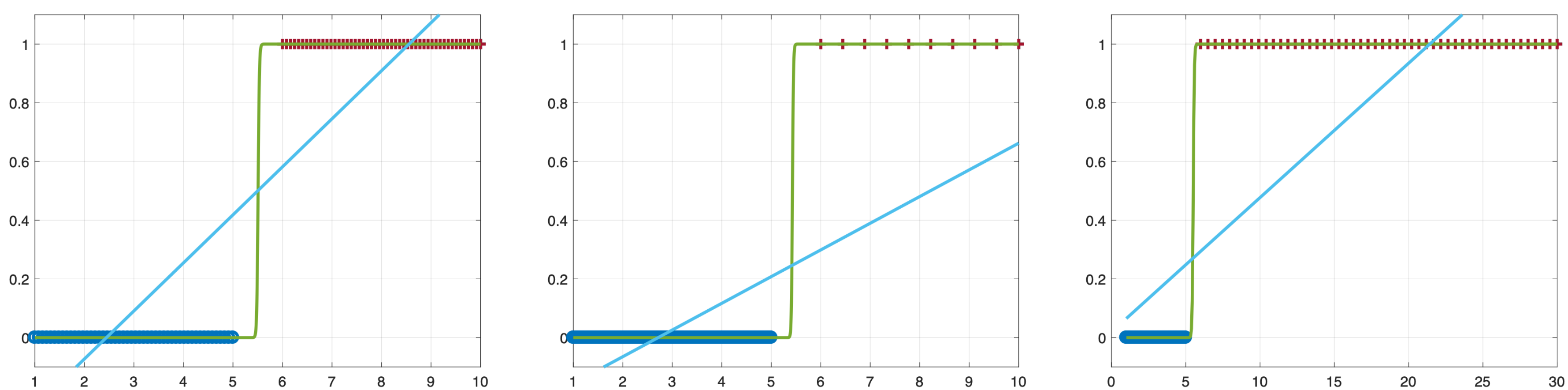
- which is exactly the anti-logarithm of **Bernoulli distribution**
  - *RECALL: **f_θ(x)** is the probability that **y = 1***

$$\ell(y; \theta, \mathrm{x}) = \Big(f_\theta(\mathrm{x})\Big)^y \Big(1 - f_\theta(\mathrm{x})\Big)^{1-y}$$

Logistic regression vs linear regression

- Logistic regression is meant for classification
  - *don't get confused by the term "regression" in its name !!!*

[Balanced classes](#) [Unbalanced classes](#) [Non-uniform classes](#)



Quiz

- In logistic regression, the cost function that computes the distance between **f_θ(x)** and **y** for an example **(x,y)** is

$$C\Big(f_\theta(\mathrm{x}), y\Big) = \begin{cases} -\log\left(f_\theta(\mathrm{x})\right) & \text{if } \ y = 1 \\ -\log\left(1 - f_\theta(\mathrm{x})\right) & \text{if } \ y = 0 \end{cases}$$

Which of the following are true? Check all that apply.

1) *If **f_θ(x) = y**, then **C(f_θ(x),y) = 0** both for **y = 0** and **y = 1**.*

2) *If **y = 0**, then **C(f_θ(x),y) ➔ 0** as **f_θ(x) ➔ 1**.*

3) *If **y = 0**, then **C(f_θ(x),y) ➔ 0** as **f_θ(x) ➔ 0**.*

4) *Regardless of whether **y = 0** or **y = 1**, if **f_θ(x) = 0.5**, then **C(f_θ(x),y) > 0**.*

What we have seen so far…

- Key ingredients of logistic regression
  - *Training data* → *Vector inputs — Binary outputs*

$$\left(\mathrm{x}^{(n)}, y^{(n)}\right) \in \mathbb{R}^Q \times \{0,1\} \qquad n = 1,\dots,N$$

  - *Prediction* → *Logistic model*

$$f_\theta(\mathrm{x}) = \frac{1}{1+\exp(-\theta^\top \mathrm{x})}$$

  - *Learning* → *Logarithmic cost function*

$$J(\theta) = \sum_{n=1}^{N} -y^{(n)} \log\left(f_\theta(\mathrm{x}^{(n)})\right) - (1-y^{(n)}) \log\left(1 - f_\theta(\mathrm{x}^{(n)})\right)$$

Performance evaluation

Confusion matrix

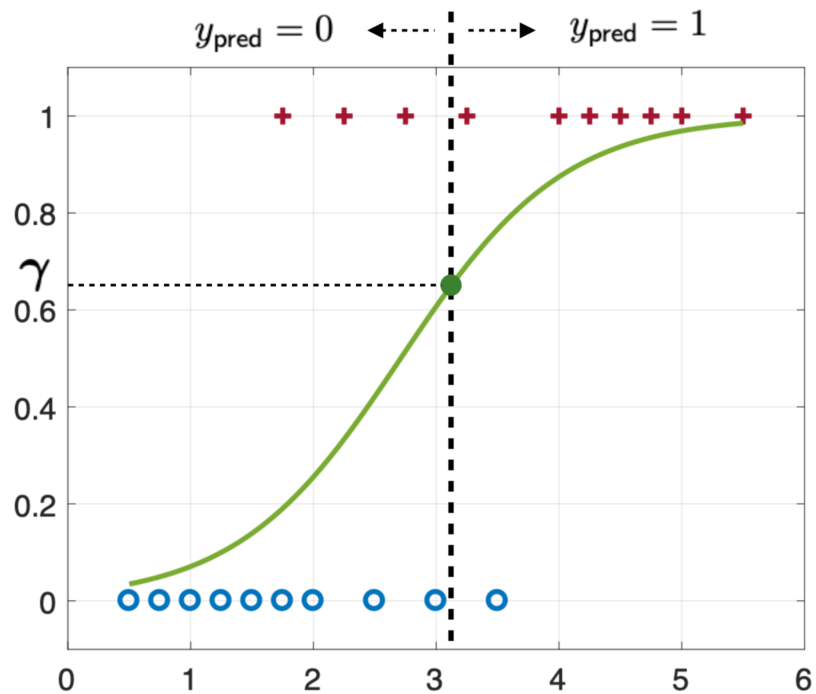
Performance scores

ROC curve

Thresholding

- Logistic model predicts a **probability** → **0 < f_θ(x) < 1**
  - *We can obtain a binary classifier by thresholding*

$$y_{\text{pred}} = \begin{cases} 1 & \text{if} \;\; f_\theta(\mathrm{x}) \geq \gamma \\ 0 & \text{if} \;\; f_\theta(\mathrm{x}) < \gamma \end{cases}$$



Confusion matrix (1/2)

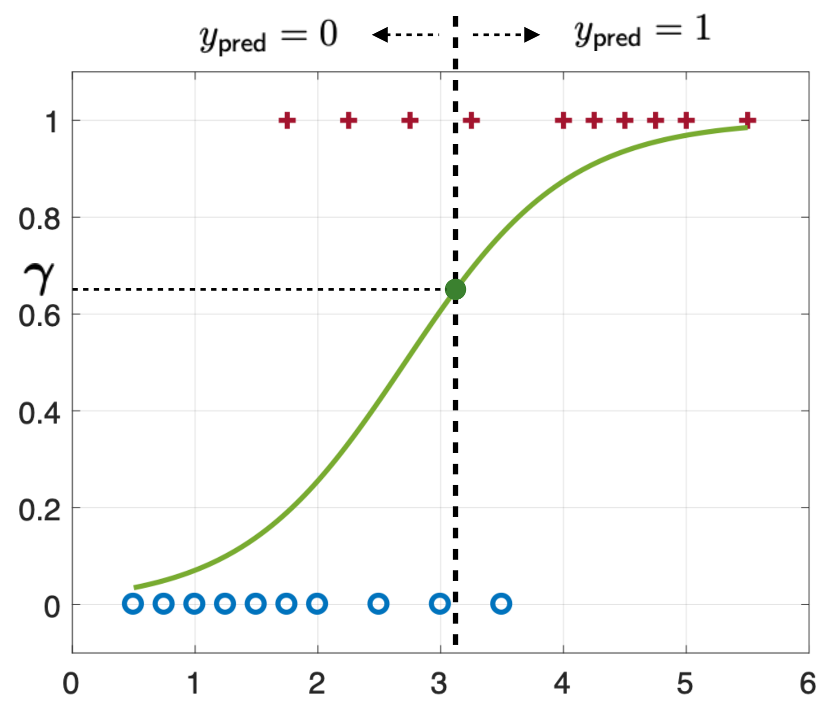
- A binary classifier can make two types of errors
  - *it is useful to count the errors occurring on the training/test data*
  - *this information can be conveniently displayed in a* ***confusion matrix***

		Actual class (y_{true})	
		0	**1**
Predicted class (y_{pred})	**0**	**True negative**	**False negative**
	1	**False positive**	**True positive**

Confusion matrix (2/2)

- EXAMPLE. Confusion matrix for the classifier below.

		Actual class (y_{true})	
		0	1
Predicted class (y_{pred})	0	9 (TN)	3 (FN)
	1	1 (FP)	7 (TP)



Overall performance (1/2)

- **Accuracy** measures the overall performance
  - *Fraction of examples that are correctly classified*

$$\text{accuracy} = \frac{\text{true positive} + \text{true negative}}{\text{total number of examples}}$$

		Actual class	
		0	1
Predicted class	0	9 (TN)	3 (FN)
	1	1 (FP)	7 (TP)

$$\text{accuracy} = \frac{9+7}{9+3+1+7} = \frac{16}{20} = 0.8$$

Overall performance (2/2)

- Accuracy is **meaningless** when classes are unbalanced

Balanced classes

$$\text{accuracy} = \frac{9+7}{9+3+1+7} = \frac{16}{20} = 0.8$$

		Actual class 0	Actual class 1
Predicted class	0	9 (TN)	3 (FN)
	1	1 (FP)	7 (TP)
		N=10	P=10

Unbalanced classes

$$\text{accuracy} = \frac{16+0}{16+3+1+0} = \frac{16}{20} = 0.8$$

		Actual class 0	Actual class 1
Predicted class	0	16 (TN)	3 (FN)
	1	1 (FP)	0 (TP)
		N=17	P=3

Class-specific scores (1/2)

- **Sensitivity** measures the true positive rate
  - *Fraction of **positive (y_{true} = 1)** correctly classified **as such (y_{pred} = 1)***

$$\text{sensitivity} = \frac{\text{true positive}}{\text{actual positive}} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}$$

		Actual class	
		0	1
Predicted class	0	9 (TN)	3 (FN)
	1	1 (FP)	7 (TP)

$$\text{sensitivity} = \frac{7}{7+3} = \frac{7}{10} = 0.7$$

Class-specific scores (2/2)

- **Specificity** measures the true negative rate
  - *Fraction of **negative (y_{true} = 0)** correctly classified **as such (y_{pred} = 0)***

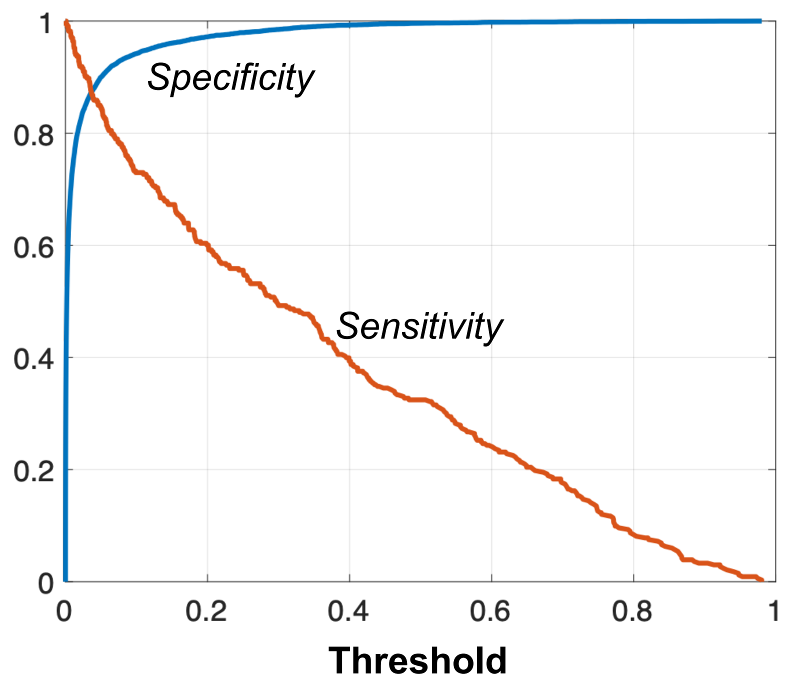
$$\text{specificity} = \frac{\text{true negative}}{\text{actual negative}} = \frac{\text{true negative}}{\text{true negative} + \text{false positive}}$$

		Actual class	
		0	1
Predicted class	0	9 (TN)	3 (FN)
	1	1 (FP)	7 (TP)

$$\text{specificity} = \frac{9}{9+1} = \frac{9}{10} = 0.9$$

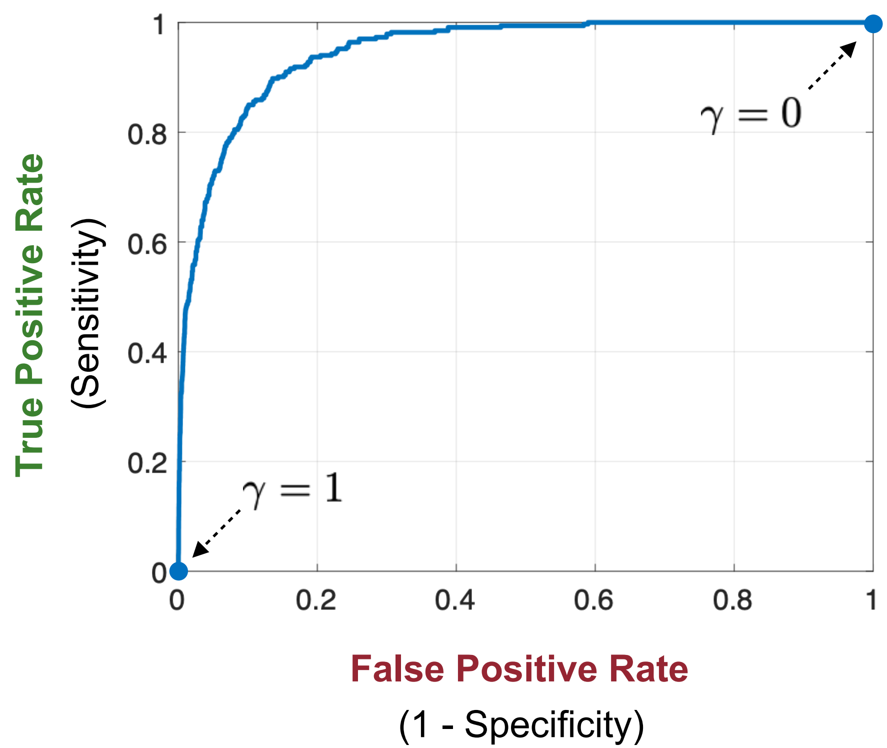
ROC curve (1/2)

- Class-specific scores are controlled by the threshold
  - ***High threshold*** **(γ > 0.5)** ➔ *high specificity, low sensitivity*
  - ***Low threshold*** **(γ < 0.5)** ➔ *low specificity, high sensitivity*



ROC curve (2/2)

- **ROC curve** plots the scores simultaneously for all $\boldsymbol{\gamma} \in [0,1]$
  - *Performance can be evaluated with the* ***area under curve***



Quiz

- A number of patients take a diagnostic test, and the results are reported in the confusion table given below. What is the sensitivity and specificity for this test?

1) ***Sensitivity***: *0.90* — ***Specificity***: *0.50*

2) ***Sensitivity***: *0.10* — ***Specificity***: *0.50*

3) ***Sensitivity***: *0.50* — ***Specificity***: *0.90*

4) ***Sensitivity***: *0.10* — ***Specificity***: *0.90*

		Actual class	
		0	**1**
Predicted	**0**	**100 (TN)**	**20 (FN)**
	1	**100 (FP)**	**180 (TP)**
		Specificity	Sensitivity

What we have seen so far…

- Logistic model makes a binary decision by thresholding

$$y_{\text{pred}} = \begin{cases} 1 & \text{if } f_\theta(\mathrm{x}) \geq \gamma \\ 0 & \text{if } f_\theta(\mathrm{x}) < \gamma \end{cases}$$

- Performance is evaluated with various scores
  - *Accuracy, Sensitivity (true positive), Specificity (true negative)*

- Comparisons can be made through ROC curves
  - *True positive (sensitivity)* ***vs*** *False positive (1-specificity)*

Decision boundary

No-prior threshold

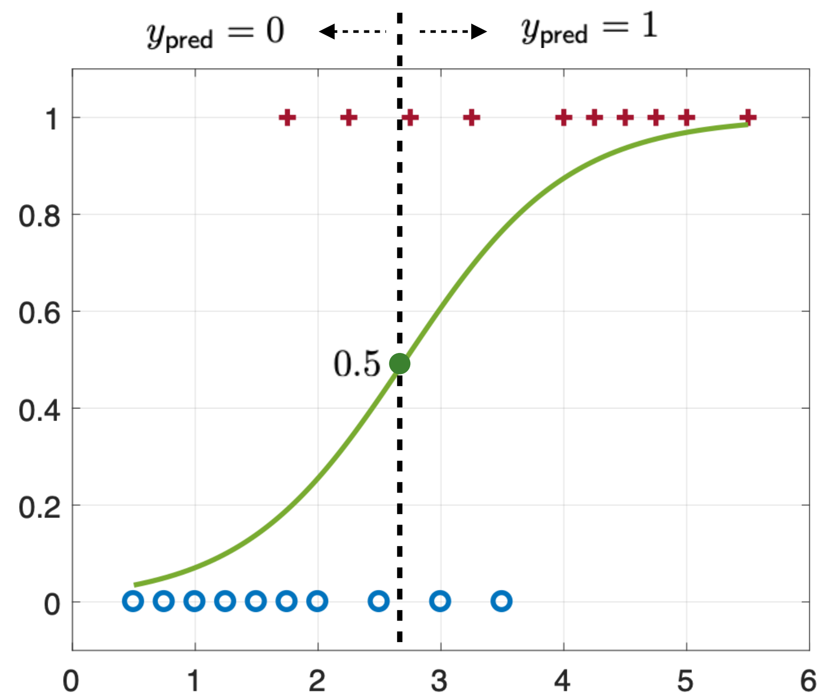
Decision boundary

Feature expansion

No-prior threshold

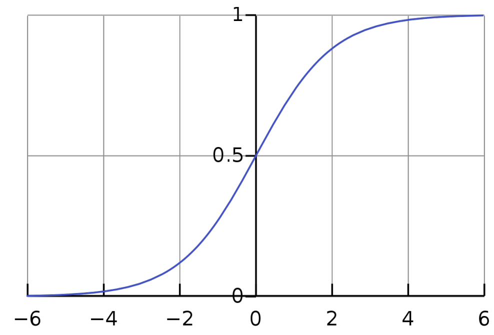
- Let's focus on the particular choice $\boldsymbol{\gamma}$ **= 0.5**
  - *this is a reasonable choice when* ***no prior knowledge*** *is available*
  - *we are simply selecting the most probable class*

$$y_{\text{pred}} = \begin{cases} 1 & \text{if } \; f_\theta(\mathrm{x}) \geq 0.5 \\ 0 & \text{if } \; f_\theta(\mathrm{x}) < 0.5 \end{cases}$$



Decision boundary (1/2)

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



- It is possible to show that

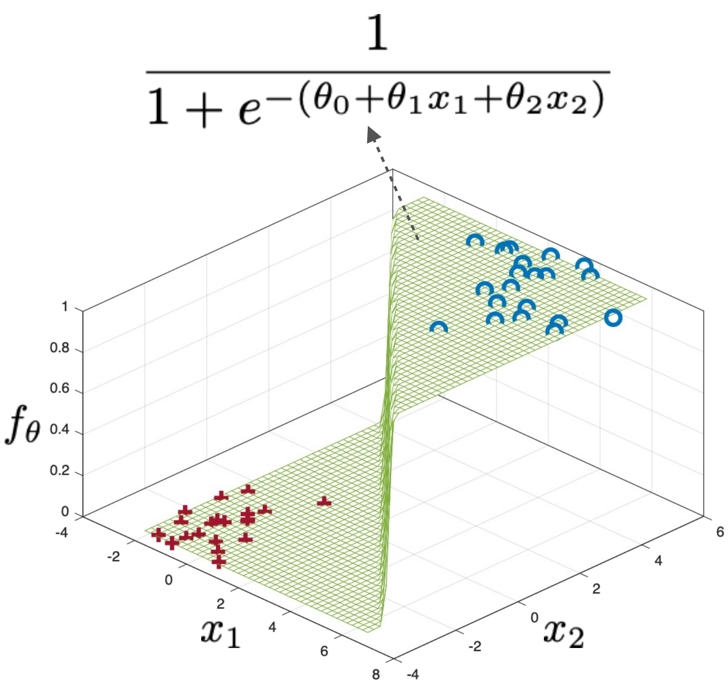
$$f_\theta(\mathrm{x}) = g(\theta^\top \mathrm{x}) \geq 0.5 \qquad \Leftrightarrow \qquad \theta^\top \mathrm{x} \geq 0$$
$$f_\theta(\mathrm{x}) = g(\theta^\top \mathrm{x}) < 0.5 \qquad \Leftrightarrow \qquad \theta^\top \mathrm{x} < 0$$

- Hence, thresholding by $\boldsymbol{\gamma}$ **= 0.5** is equivalent to

$$y_{\text{pred}} = \begin{cases} 1 & \text{if } \theta^\top \mathrm{x} \geq 0 \\ 0 & \text{if } \theta^\top \mathrm{x} < 0 \end{cases}$$

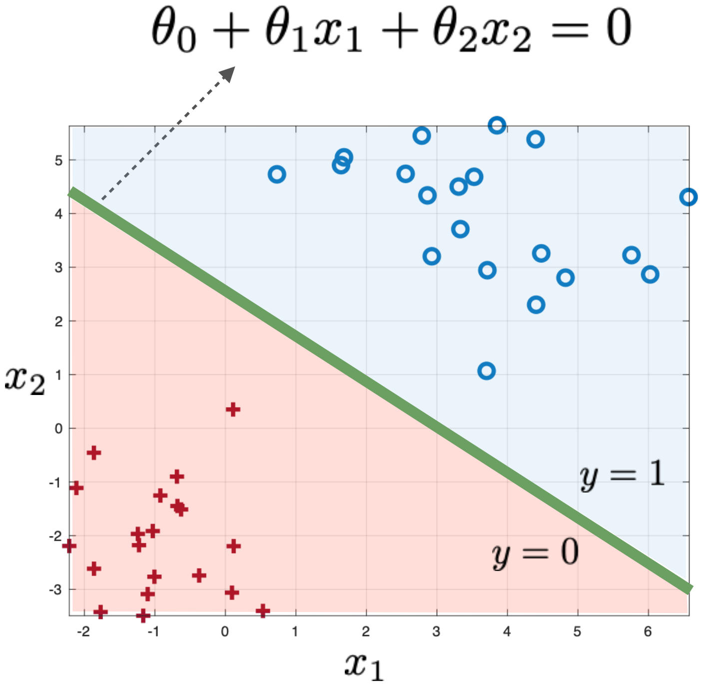
Decision boundary (2/2)

- Logistic regression is a **linear classifier**
  - *the feature space is split in two regions by an hyperplane*



Separating hyperplane

defined with γ = 0.5



Feature expansion (1/3)

- How can we learn a nonlinear classifier ?

  1) Transform the input into new variables…

$$\phi(\mathrm{x}) = \begin{bmatrix} \phi_1(\mathrm{x}) \\ \vdots \\ \phi_M(\mathrm{x}) \end{bmatrix} \in \mathbb{R}^M \qquad \text{with} \quad \phi_m \colon \mathbb{R}^Q \to \mathbb{R}$$

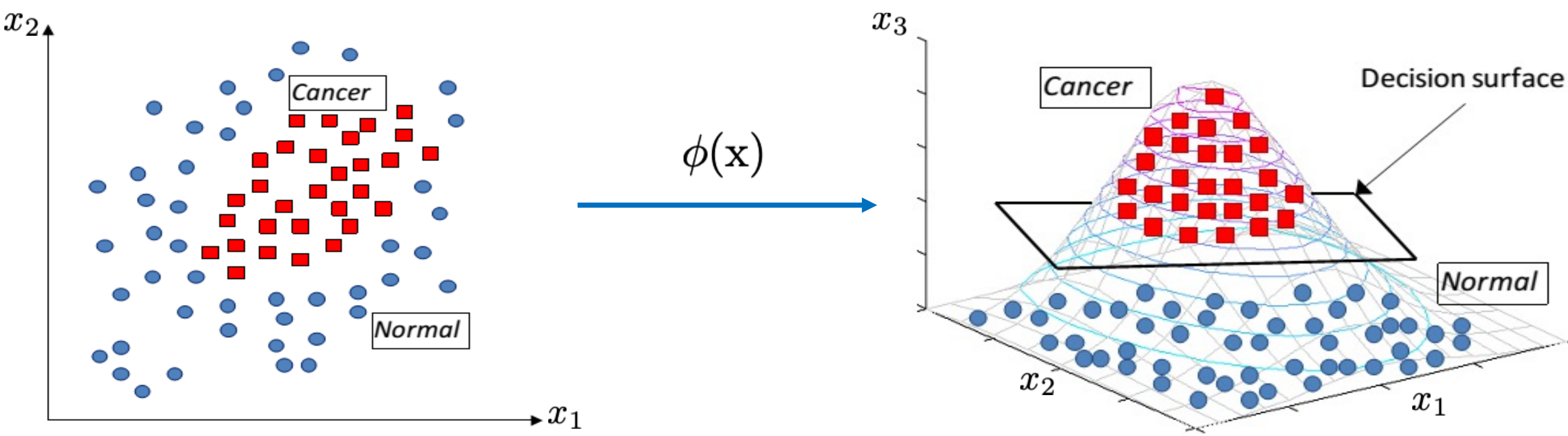
  2) … and put them in the logistic model

$$f_\theta(\mathrm{x}) = g\Big(\theta^\top \phi(\mathrm{x})\Big) = g\Big(\theta_0 + \theta_1\phi_1(\mathrm{x}) + \cdots + \theta_M\phi_M(\mathrm{x})\Big)$$

Feature expansion (2/3)

- This amounts to
  - *mapping the data into a higher dimensional space*
  - *fitting the transformed data with a linear model*

Original samples — Transformed samples



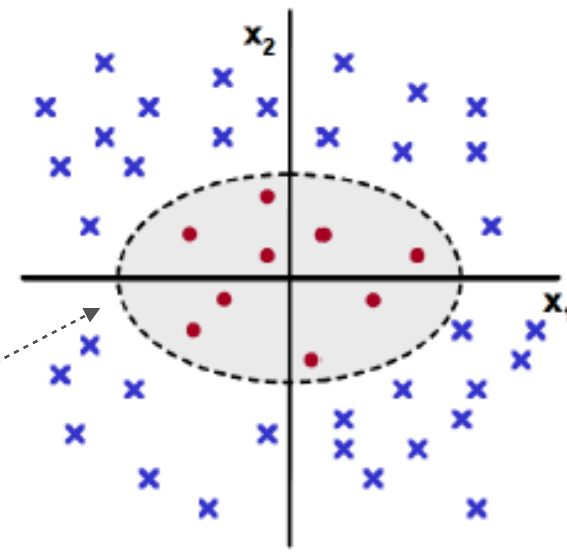
Feature expansion (3/3)

- EXAMPLE. Polynomial mapping
  - *add extra features by raising each of the original ones to a power*
  - *decision boundary is now a polynomial equation*

$$f_\theta(\mathrm{x}) = g\big(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2\big)$$

$$\Downarrow$$

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 = 0$$

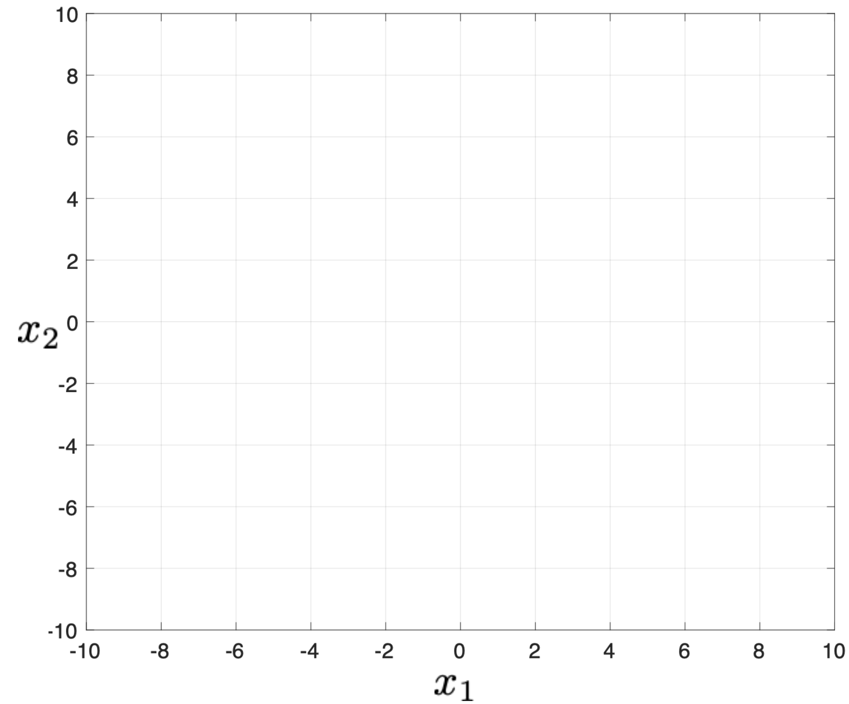


Quiz (1/2)

- Suppose you trained a logistic regression classifier with two features: $\mathbf{f_\theta(x) = g(\theta_0 + \theta_1\, x_1 + \theta_2\, x_2)}$.

1) Draw the decision boundary for the classifiers given by

- $\boldsymbol{\theta = [\theta_0, \theta_1, \theta_2] = [-6, 1, 0]}$
- $\boldsymbol{\theta = [\theta_0, \theta_1, \theta_2] = [0, 1, 2]}$
- $\boldsymbol{\theta = [\theta_0, \theta_1, \theta_2] = [-3, 1, 1]}$



2) Once the parameters $\boldsymbol{\theta}$ *have been learned from the training data, do you still need such data to draw the decision boundary of a logistic classifier ?*

Quiz (2/2)

- Suppose you have the following training set, and fit a logistic regression $\mathbf{f_\theta(x) = g(\theta_0 + \theta_1\, x_1 + \theta_2\, x_2)}$.
- Which of the following are true? Check all that apply.

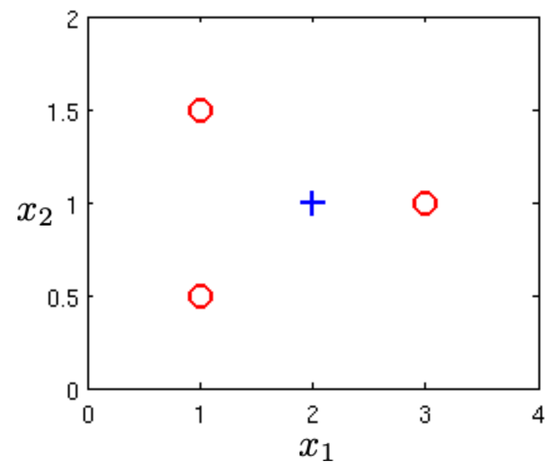
x_1	x_2	y
1	0.5	0
1	1.5	0
2	1	1
3	1	0

1) *The positive and negative examples cannot be separated using a straight line.*

2) *Because the positive and negative examples cannot be separated using a straight line, linear regression will perform as well as logistic regression on this data.*

3) *Adding polynomial features could improve data fitting:*
$$\boldsymbol{f_\theta(x) = g(\theta_0 + \theta_1\, x_1 + \theta_2\, x_2 + \theta_3\, x_1\, x_2 + \theta_4\, x_1^2 + \theta_5\, x_2^2)}.$$

4) *Logistic regression is a linear classifier, and thus it can only separate the feature space by an hyperplane.*



What we have seen so far…

- Logistic regression is a linear classifier

$$y_{\text{pred}} = \begin{cases} 1 & \text{if } \theta^\top \mathrm{x} \geq 0 \\ 0 & \text{if } \theta^\top \mathrm{x} < 0 \end{cases} \qquad (\text{assuming } \gamma{=}0.5)$$

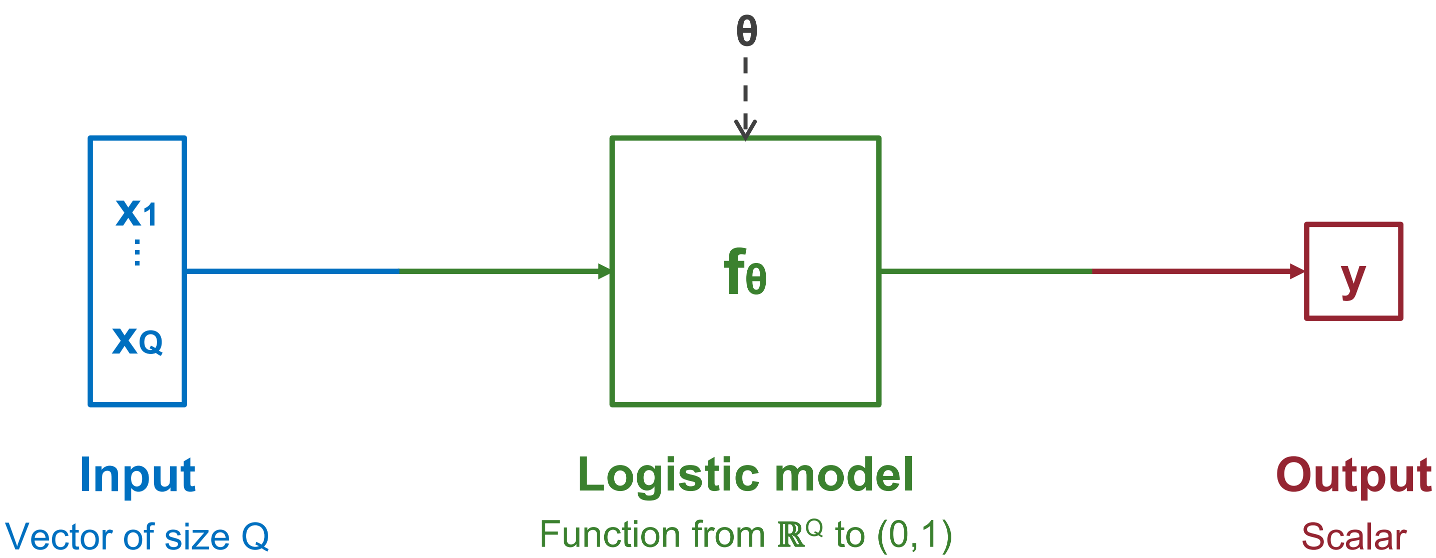
- Feature expansion can make it a nonlinear classifier

$$y_{\text{pred}} = \begin{cases} 1 & \text{if } \theta^\top \phi(\mathrm{x}) \geq 0 \\ 0 & \text{if } \theta^\top \phi(\mathrm{x}) < 0 \end{cases} \qquad (\text{assuming } \gamma{=}0.5)$$

Multiclass classification

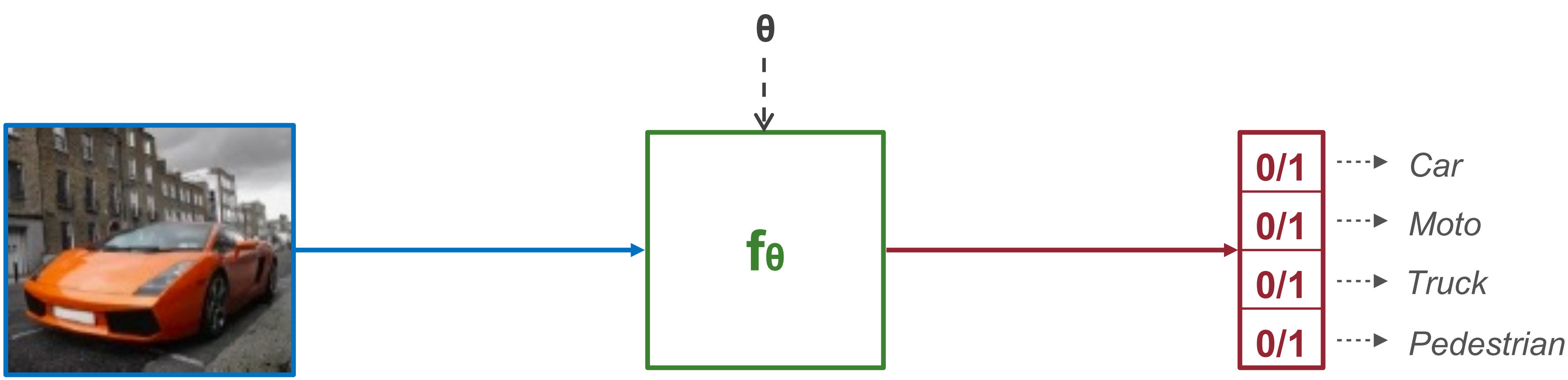
Single output

- The logistic model outputs a **single number**
  - **Classification →** Two classes



Multiple outputs

- How to handle **multiclass classification** ?
  - The classifier needs to predict multiple outputs (one per class)



Input

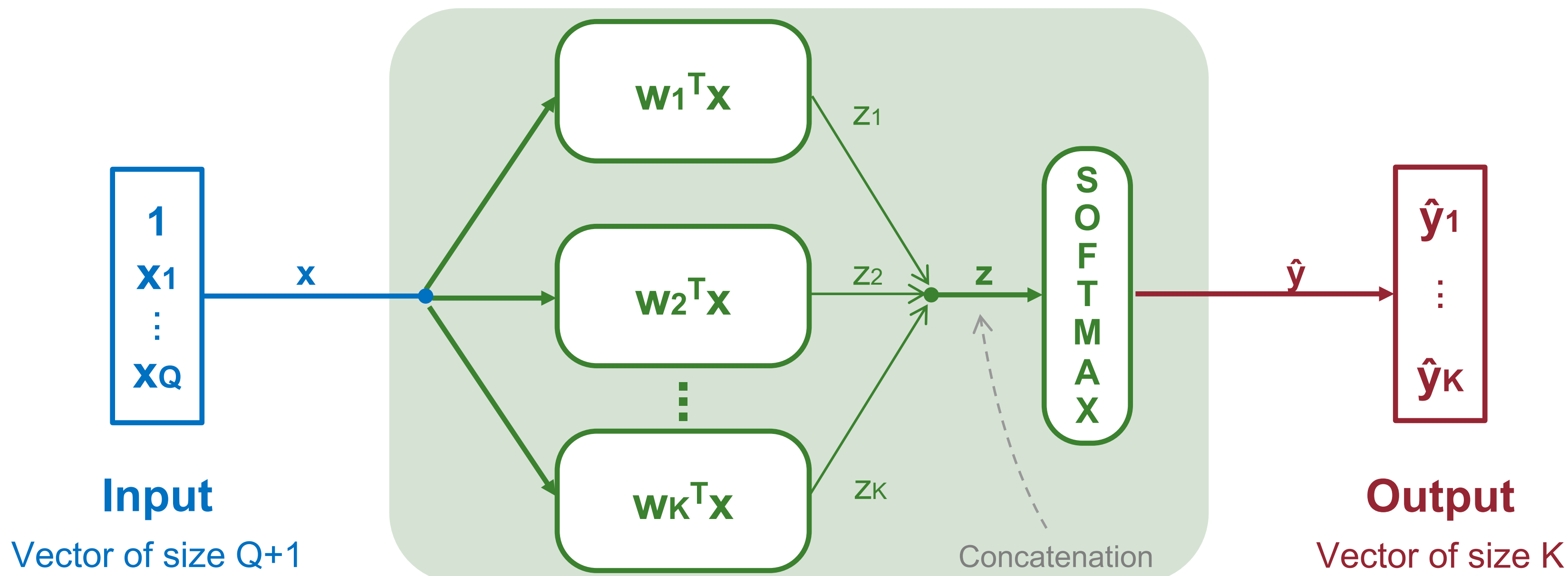
Classifier

Function from $\mathbb{R}^Q$ to $\{1,\ldots,K\}$

Output

Multiclass logistic regression (1/4)

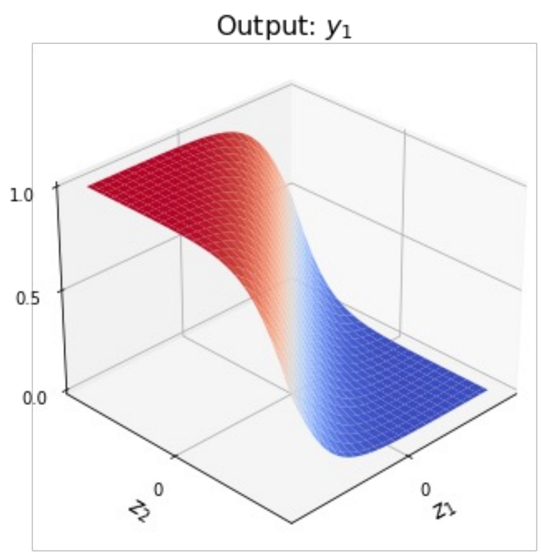
- **Multiclass logistic regression**
  - The vector input is supplied to multiple linear models
  - The results of these models are concatenated into a vector
  - The vector is transformed by the “softmax” and sent to the output

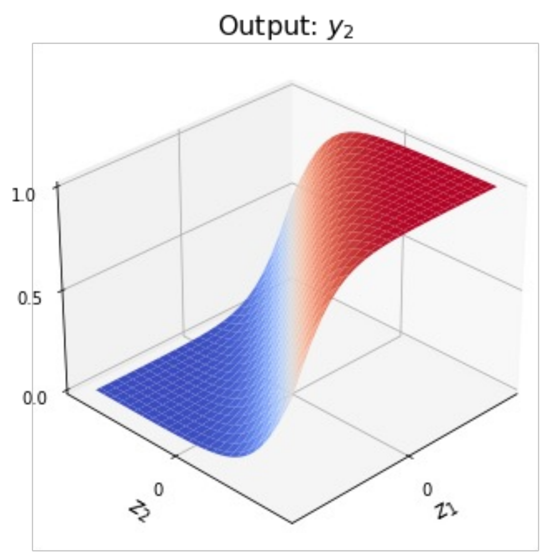


Multiclass logistic regression (2/4)

- **Softmax**
  - A vector is transformed to have positive elements that sum to one
  - Generalization of the sigmoid to multiple dimensions
  - Smooth approximation of the “argmax” operation

$$\mathbf{g}(\mathbf{z}) = \begin{bmatrix} \dfrac{e^{z_1}}{e^{z_1}+\cdots+e^{z_K}} \\ \dfrac{e^{z_2}}{e^{z_1}+\cdots+e^{z_K}} \\ \vdots \\ \dfrac{e^{z_K}}{e^{z_1}+\cdots+e^{z_K}} \end{bmatrix}$$





Multiclass logistic regression (3/4)

- **Training data**
  - Vector input — Vector output

$$\mathcal{S}_{\text{train}} = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) \in \mathbb{R}^Q \times \{0,1\}^K \mid n = 1, \dots, N\}$$

  - Output vectors must be **one-hot encoded**

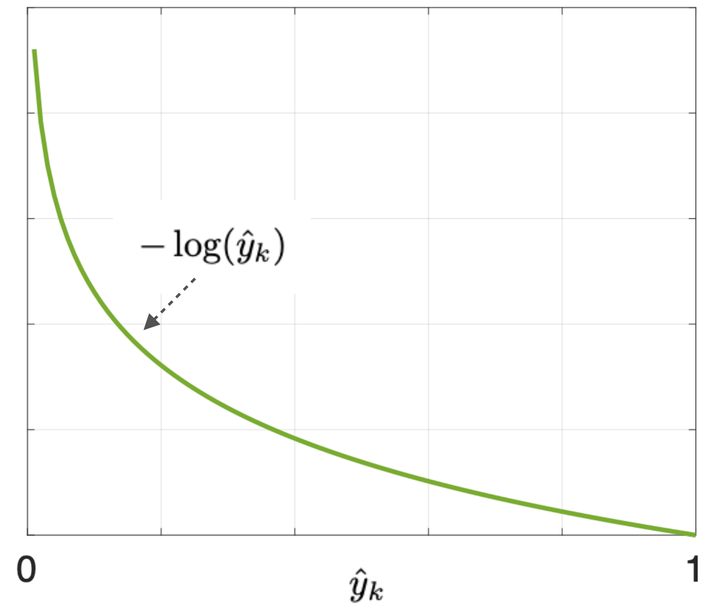
$$\mathbf{y}_{\text{class 1}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{y}_{\text{class 2}} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \qquad \cdots \qquad \mathbf{y}_{\text{class K}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Multiclass logistic regression (4/4)

- **Loss function →** Cross entropy

$$\mathcal{E}(\hat{\mathbf{y}}, \mathbf{y}) = \begin{cases} -\log\left(\hat{y}_1\right) & \text{if } y_1 = 1 \\ -\log\left(\hat{y}_2\right) & \text{if } y_2 = 1 \\ \vdots & \\ -\log\left(\hat{y}_K\right) & \text{if } y_K = 1 \end{cases}$$

Only one is selected ← - - - *One-hot encoding*



Training

- How to select the **right values** for the parameters?
  - Minimize the mean error of prediction on the training data

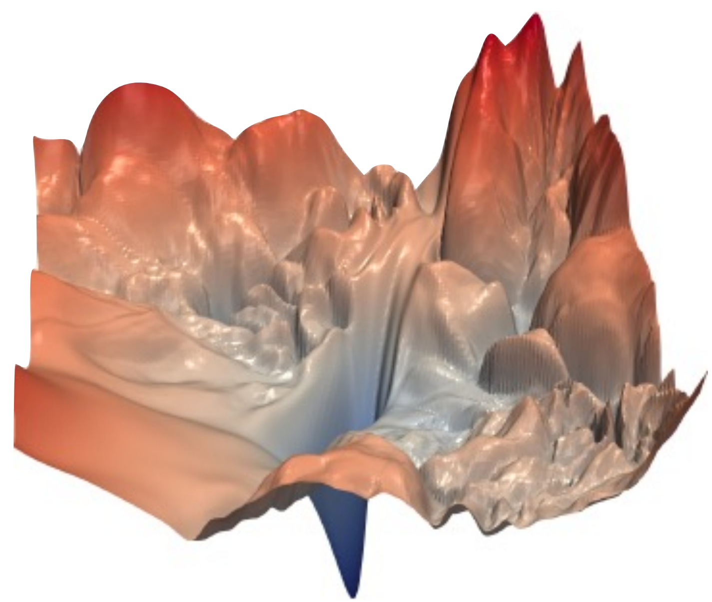
$$\underset{\theta}{\text{minimize}} \quad \frac{1}{N}\sum_{n=1}^{N} \mathcal{E}\Big(f_\theta\big(\mathbf{x}^{(n)}\big), \mathbf{y}^{(n)}\Big)$$

Parameters of the model

Output of the model

n-th sample in the training data

Cross entropy



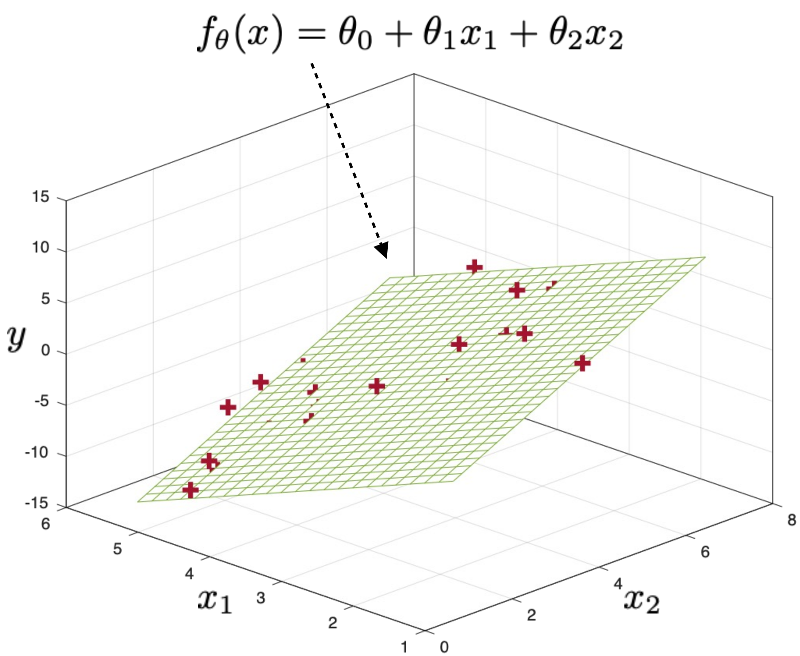
Conclusion

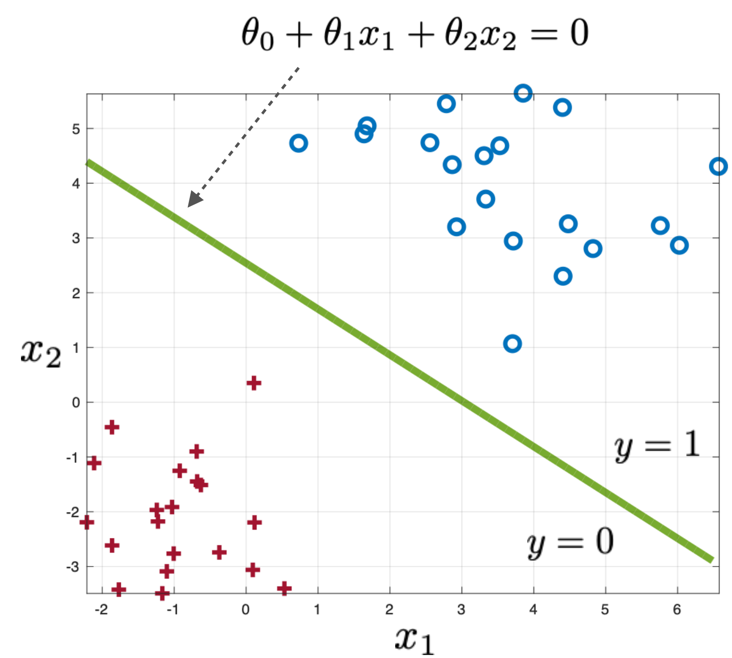
Regression vs Classification

Linear regression vs Logistic regression

Regression vs Classification

- Supervised learning can be categorized into
  - *regression* → *learning how to predict a **continuous** response*
  - *classification* → *learning how to predict a **discrete** response*





Linear regression vs Logistic regression

- Key ingredients of generalized linear models
  - *Training data*

$$(\mathrm{x}^{(n)}, y^{(n)}) \in \mathbb{R}^Q \times \mathcal{Y} \quad \begin{cases} \xrightarrow{\text{regression}} & \mathcal{Y} = \mathbb{R} \\ \xrightarrow{\text{classification}} & \mathcal{Y} = \{0, 1\} \end{cases}$$

  - *Prediction*

$$f_\theta(\mathrm{x}) = g(\theta^\top \mathrm{x}) \quad \begin{cases} \xrightarrow{\text{regression}} & g(z) = z \\ \xrightarrow{\text{classification}} & g(z) = \dfrac{1}{1 + \exp(-z)} \end{cases}$$

  - *Learning*

$$J(\theta) = \sum_{n=1}^{N} C\Big(f_\theta(\mathrm{x}^{(n)}), y^{(n)}\Big) \quad \begin{cases} \xrightarrow{\text{regression}} & \text{Squared error function} \\ \xrightarrow{\text{classification}} & \text{Logarithmic cost function} \end{cases}$$