Introduction to
Supervised Learning

Farhang Habibi

Contents

- Supervised learning
- Linear regression
- Logistic regression

Supervised Learning

- It is about predicting or identifying unknowns from known data

- This means you need sufficient quantity of data to compute some parameters from them. The parameters are learnt from data, under the machine supervision, to predict or identify some thing unknown that does not exist in your data set.

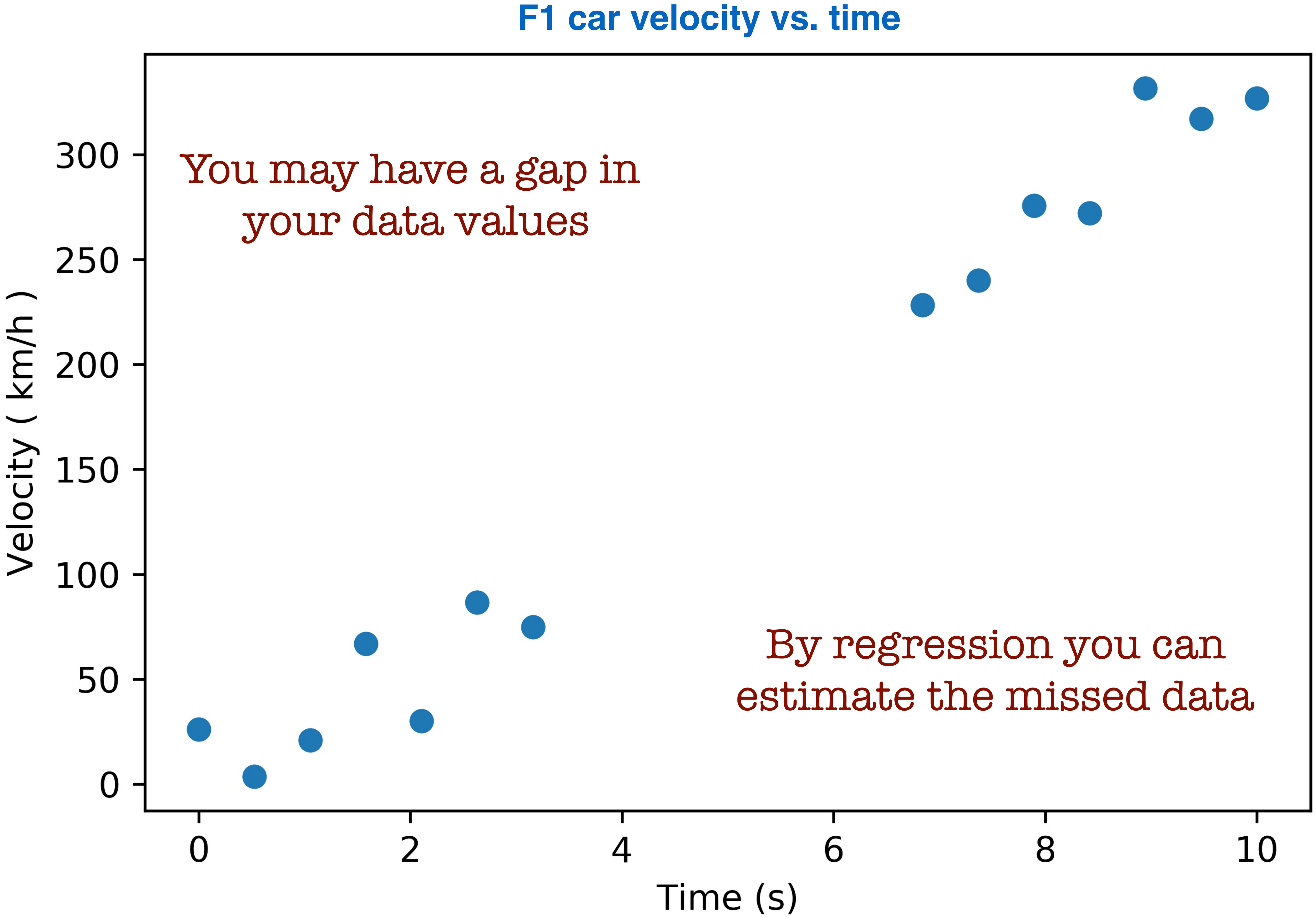
- There are 2 aspects of supervised learning :

  1- Regression (prediction for continuous variables)
  2- Classification (separating the objects with different types (classes) from each other)

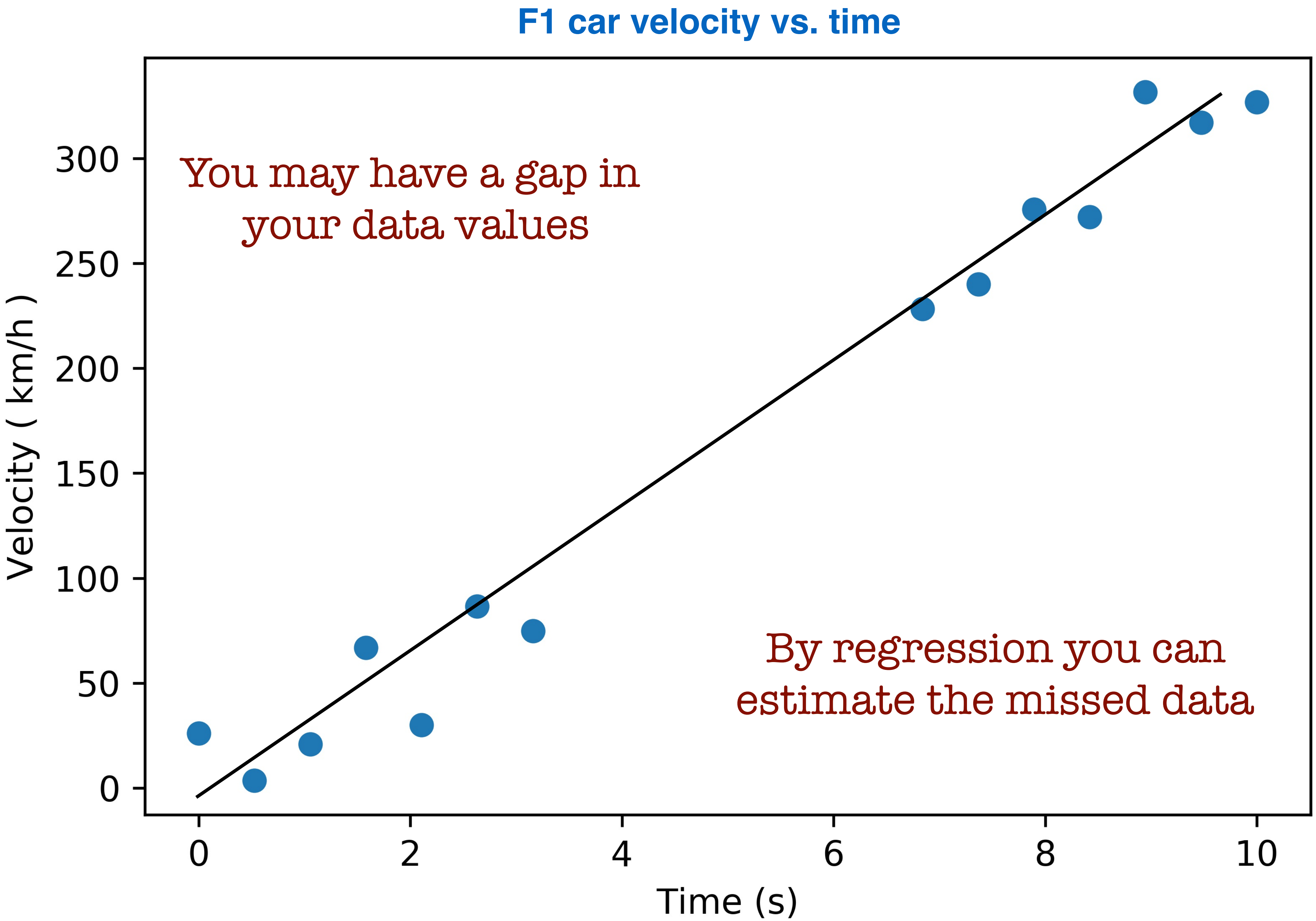
- Regression : in your data set there is one or more independent variable(s) (or features) and a dependent variable that depends on the feature(s) through a model.

- Classification : you have different classes and each class has a set of features in your data set. Same classes have similar feature values. We try to find the parameters for a mathematical process to categorise the classes through their features.

Regression Example

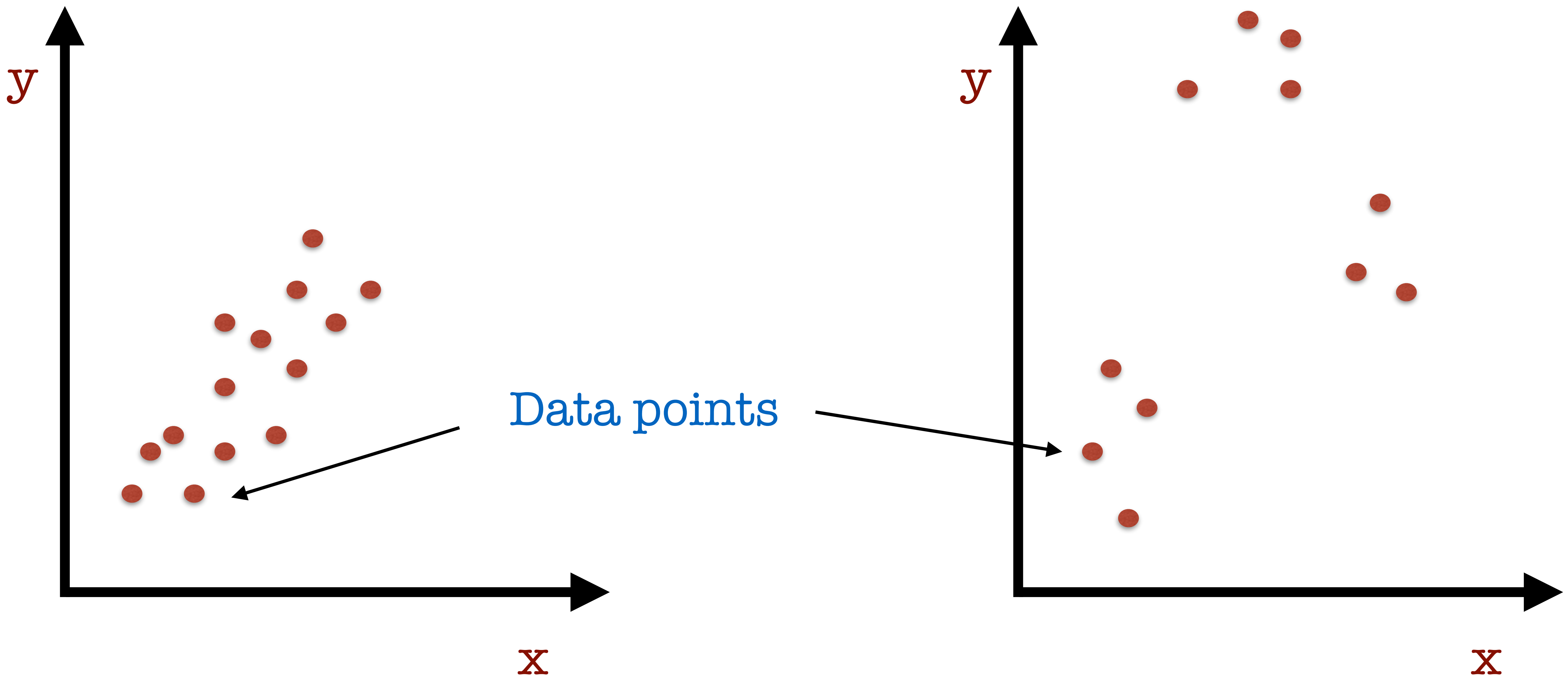


Regression Example



Regression

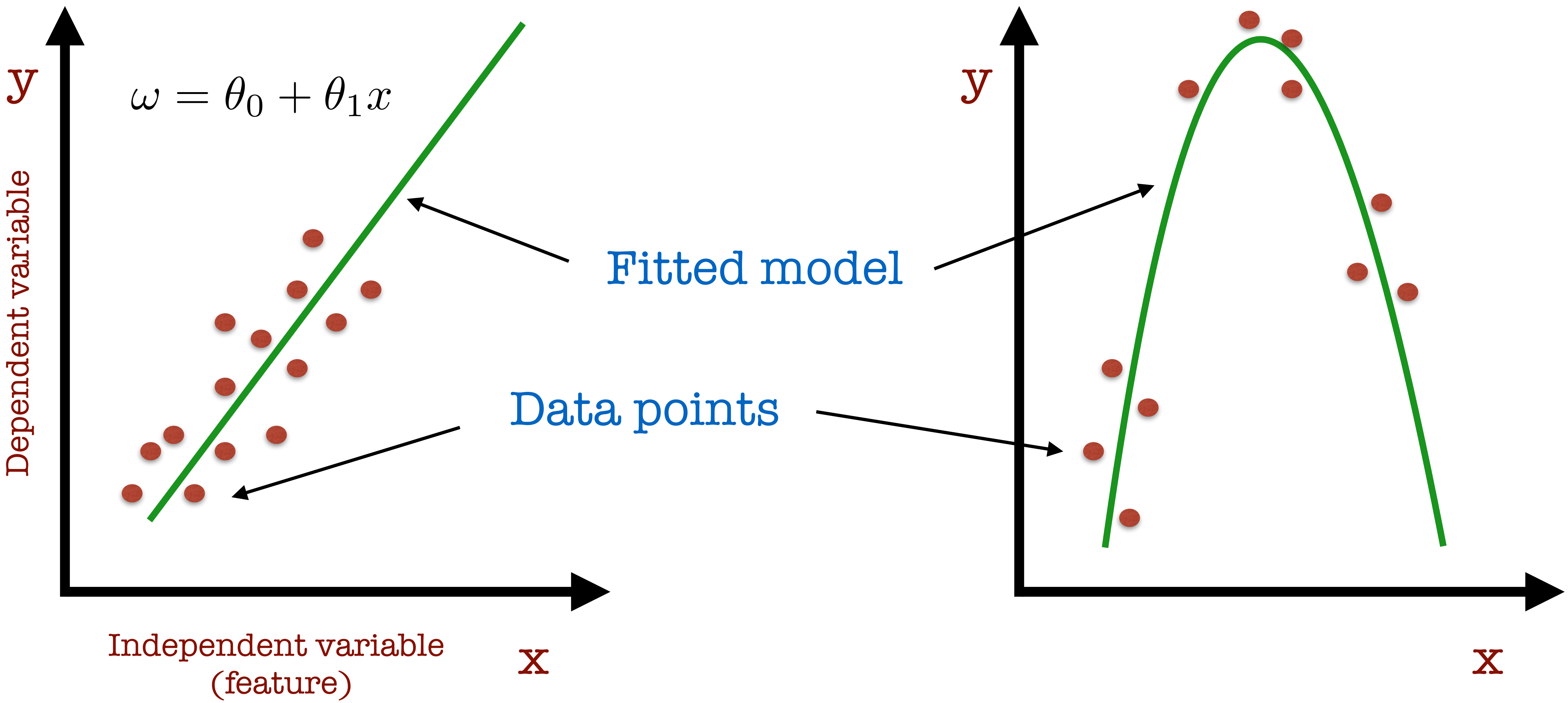
Do experiments → Data acquisition



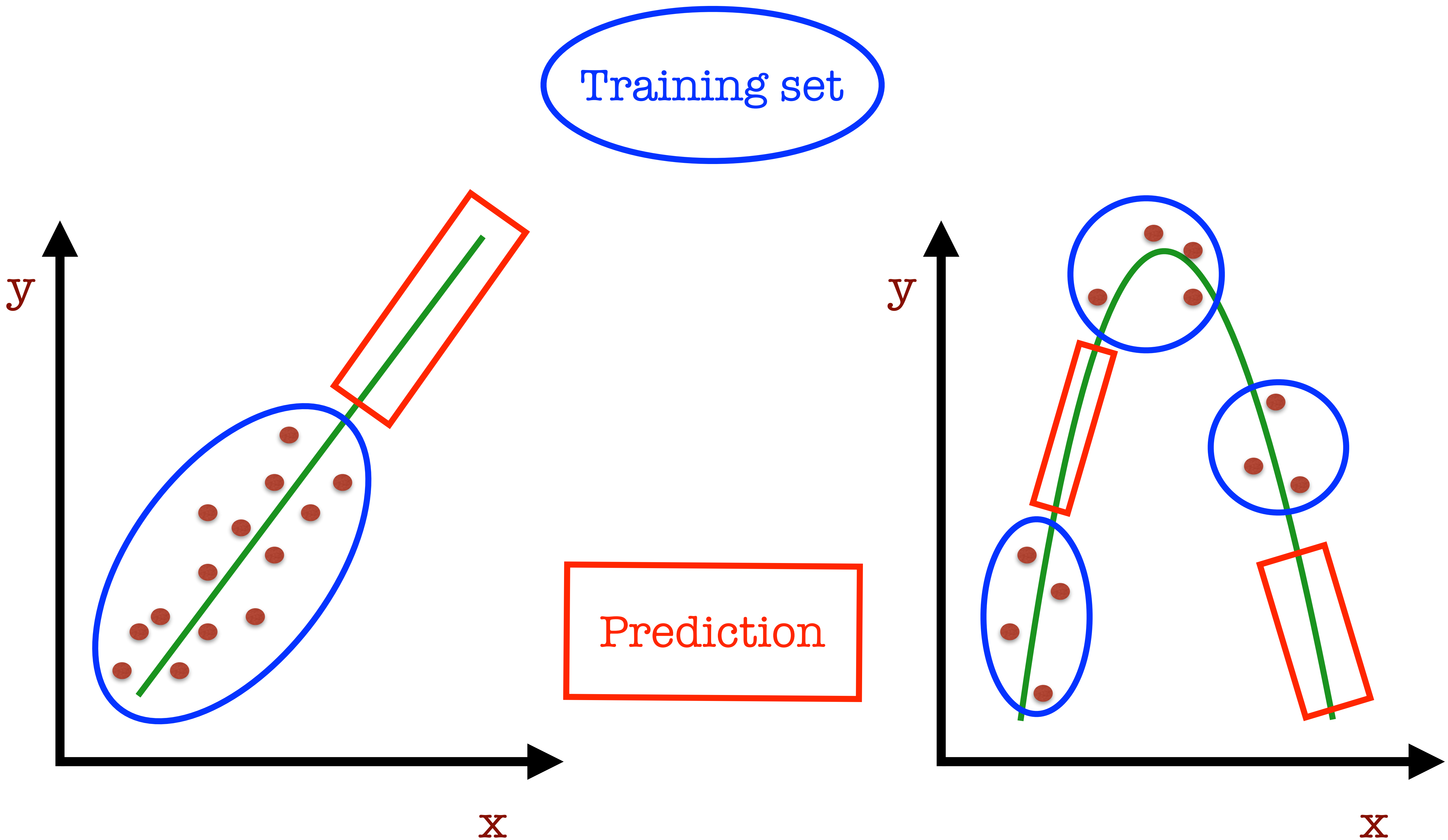
Regression

Fit a model to data to determine $\vec{\theta}$

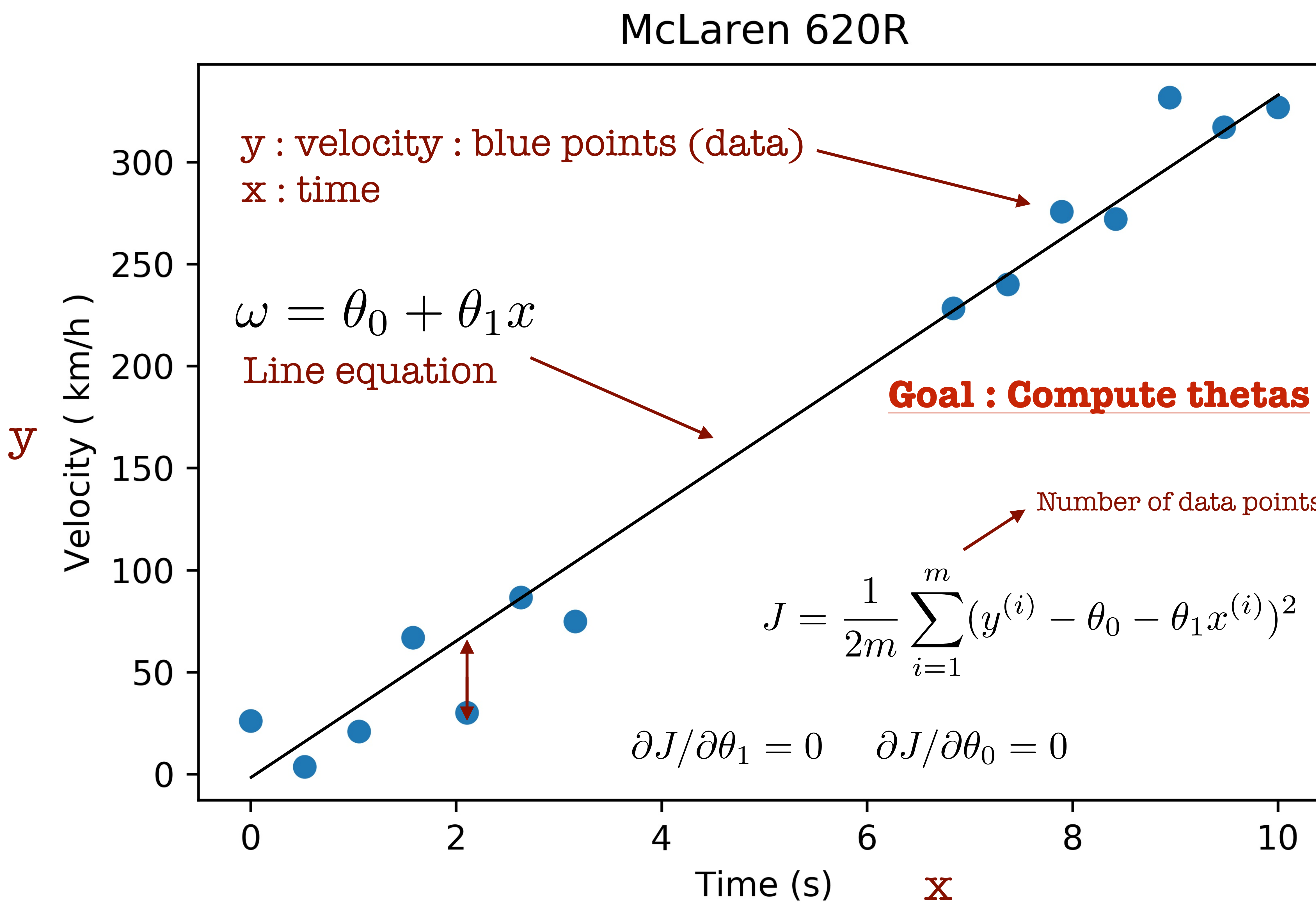
$$y = \omega(x; \vec{\theta})$$



Regression



Regression


McLaren 620R
y : velocity : blue points (data)
x : time
$\omega = \theta_0 + \theta_1 x$
Line equation
Goal : Compute thetas
Number of data points
$J = \frac{1}{2m}\sum_{i=1}^{m}(y^{(i)} - \theta_0 - \theta_1 x^{(i)})^2$
$\partial J/\partial\theta_1 = 0$
$\partial J/\partial\theta_0 = 0$
Velocity (km/h)
y
Time (s)
x
0
50
100
150
200
250
300
0
2
4
6
8
10

Regression

Linear (multiple) Regression

$$\omega(\vec{x};\vec{\theta}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$\vec{x} = (x_1, x_2, ..., x_n)$ Vector of features

$\vec{\theta} = (\theta_0, \theta_1, ..., \theta_n)$ Vector of parameters

Linear model

Cost function to be minimised to determine the parameters $\vec{\theta}$

$$J = \frac{1}{2\,m}\sum_{i=1}^{m}[\,y^{(i)} - \omega(\vec{x}^{(i)};\vec{\theta})\,]^2$$

An oval-shaped function with a single minimum with respect to $\vec{\theta}$

Regression

m measurements contain **m** number of y and **m** number of $\vec{x}$

m : number of training objects (Ntrain in the exercises)

$$X = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix} \qquad \vec{y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

Feature matrix
(matrix of independent variables)

vector of
dependent variables

Regression

$$y = \omega(\vec{x};\vec{\theta}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$J = \frac{1}{2\,m}\sum_{i=1}^{m}\left[\, y^{(i)} - \omega(\vec{x}^{(i)};\vec{\theta})\,\right]^2$$

Minimising the cost function with respect to $\vec{\theta}$

$$\vec{\theta} = (X^T\,X)^{-1}\,X^T\,\vec{y}$$

Polynomial Regression

$$y = \omega(\vec{x};\vec{\theta}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Higher degrees of x can be included by defining:

$$x_1 = x,\ x_2 = x^2,\ \ldots,\ x_n = x^n$$

Linear Regression method can be used for non-linear models as well

Dummy Variables

Categorical variables

District (X)	Price (y)
A	6000
B	5000
C	7000
B	...

Mathematical equations can not be applied on categorical variables.

6000	0	0
5000	1	0
7000	0	1
...	...	...

These variables are firstly converted to ordinal numbers.

0	6000
1	5000
2	7000
B	...

Then will be splitted to columns of dummy variables

6000	0	0	1
5000	1	0	0
7000	0	1	0
...	...	...	...

Ordinal numbers → Dummy variables

Regression

- The line is fitted to the data of the training set.
- If the training set does not have enough statistics the fitted parameters are not reliable.
- The fitted parameters should be used by a data set other than the training set to verify the accuracy of the fit.
- This independent data set is called the test set. Similars to the training set, it contains both the known features and dependent variables.
- We usually use 70%-80% of our data as the training set and the remainder as the test set.

Regression

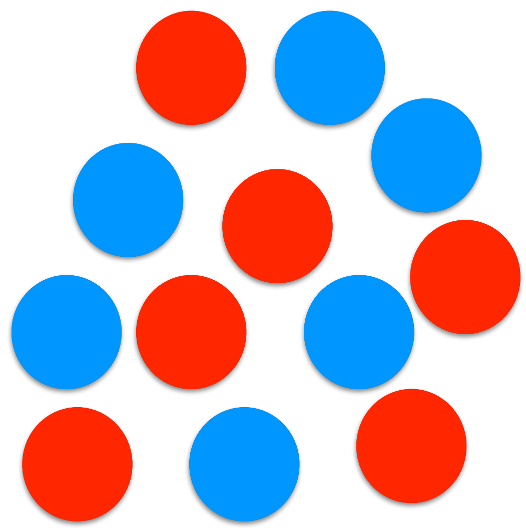
Goodness of the fit can be measured by computing the fit score

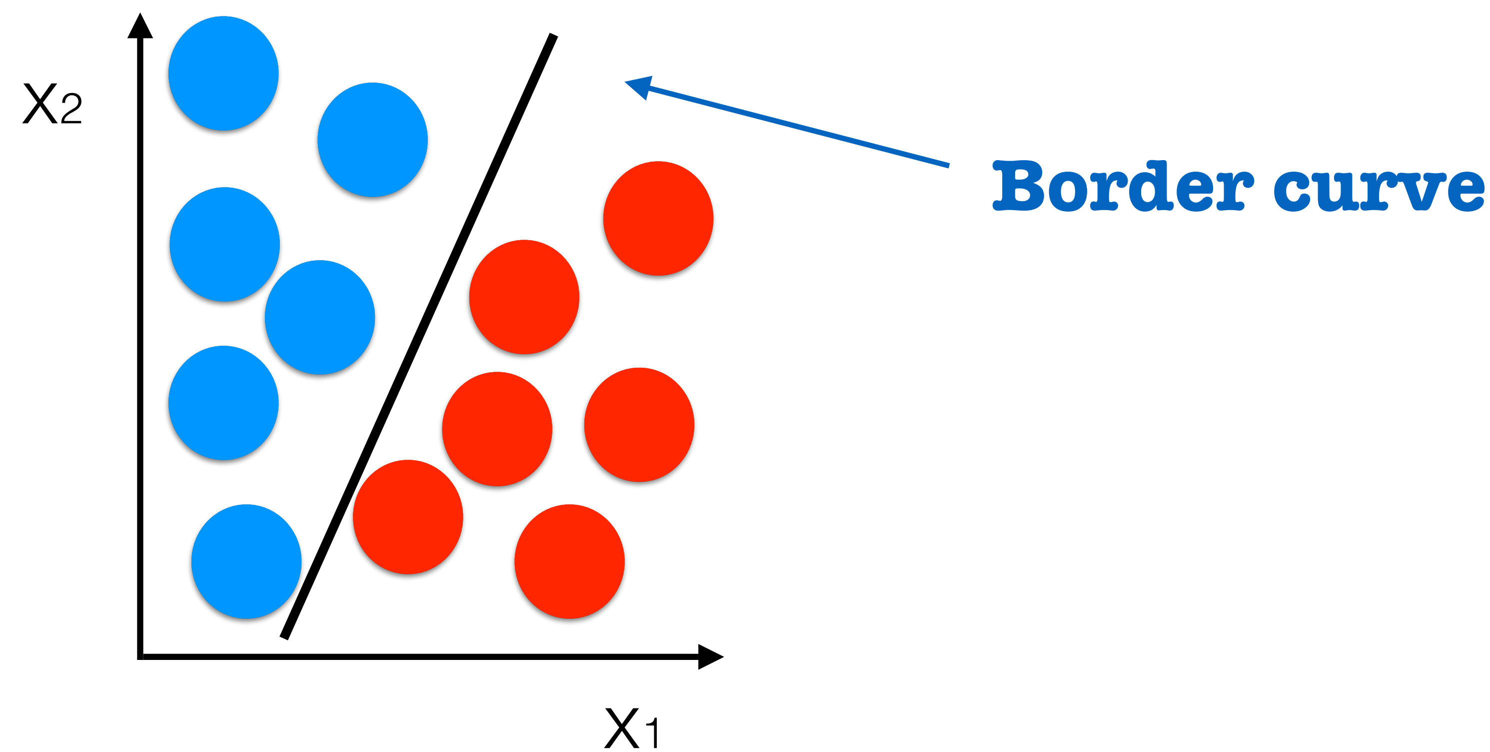
$$\text{score} = 1 - \frac{2J}{\sigma^2}$$

J is the cost function (qui-square)
sigma is the standard deviation

$$\sigma^2 = \frac{1}{m-1} \sum_{i=1}^{m} (y^{(i)} - \bar{y})^2$$

Supervised classification





Label vector

$$\vec{y} = \begin{pmatrix} y^{(1)} = 0 \\ y^{(2)} = 1 \\ y^{(3)} = 1 \\ \vdots \\ y^{(m)} = 0 \end{pmatrix}$$

Feature matrix

$$X = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$$

Supervised classification

How to classify a mixture of objects with different classes?

For objects with known classes:
Make a training set: Features + Labels (0 or 1)
Determine the parameters of the border curve

For objects with unknown classes:
Classify them (predict their labels) by
using their features + the determined parameters

Supervised classification

Logistic (Sigmoid) function

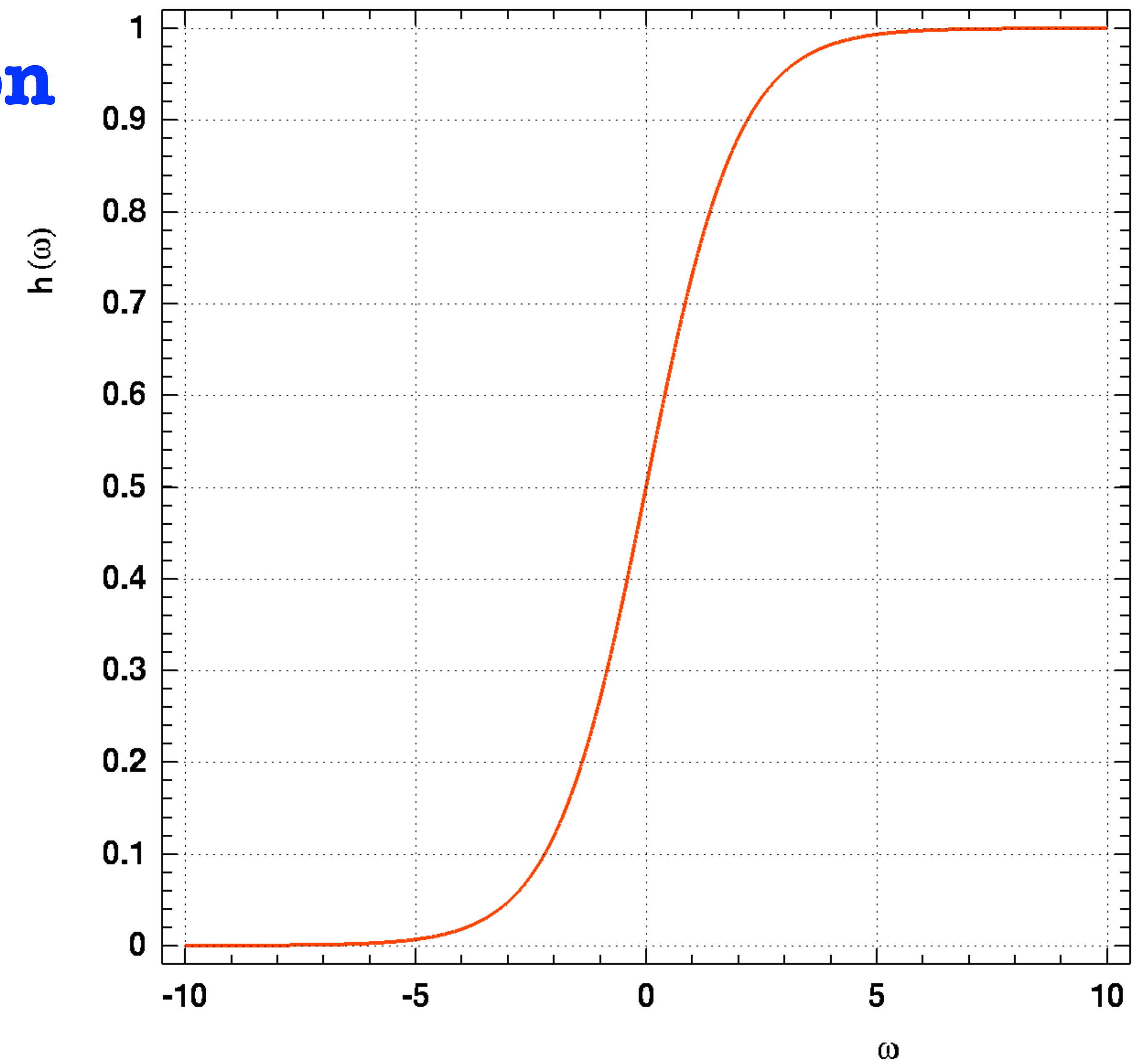
$$h(\omega) = \frac{1}{1 + e^{-\omega}}$$

$$\omega = \theta_0 + \theta_1 x$$

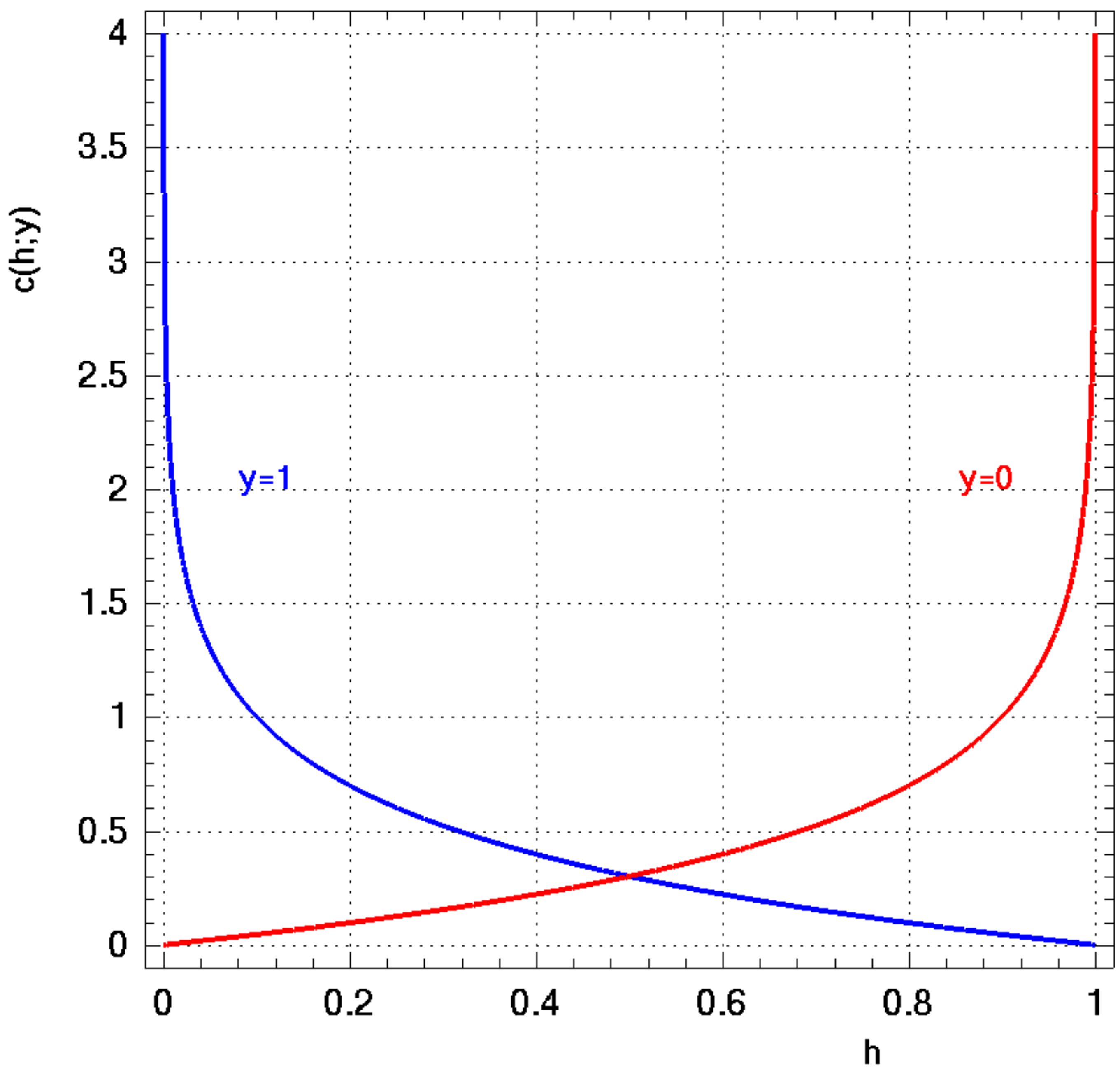
Object probability to be classified as 0 or 1

$$h(\omega = 0) = 0.5$$

Classification border line



Supervised classification



$$c(h;y) = -y\log(h) - (1-y)\log(1-h)$$

Supervised classification

$$h_\theta(\vec{x}) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n)}}$$

Hyper border surface:

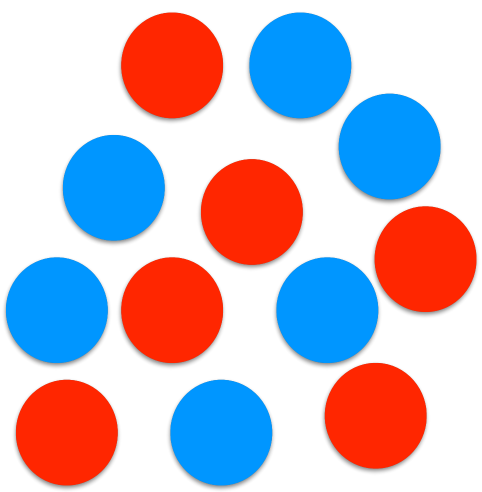
$$\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n = 0$$

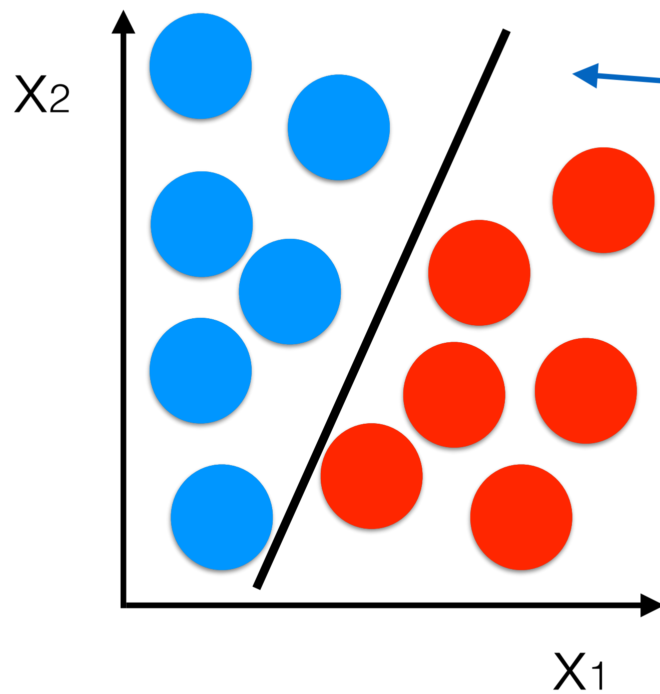
The cost function:

$$J = \frac{1}{m} \sum_{i=1}^{m} c(h_\theta(\vec{x}^{(i)}); y^{(i)})$$

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

Supervised classification





Border curve

$$\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n = 0$$

$$h_\theta(\vec{x}) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n)}}$$

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

Label vector

$$\vec{y} = \begin{pmatrix} y^{(1)} = 0 \\ y^{(2)} = 1 \\ y^{(3)} = 1 \\ \vdots \\ y^{(m)} = 0 \end{pmatrix}$$

Feature matrix

$$X = \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$$

Logistic Regression

Supervised classification

Gradient descend method

to minimise the cost function

$$\frac{\partial}{\partial \theta_j} J = 0$$

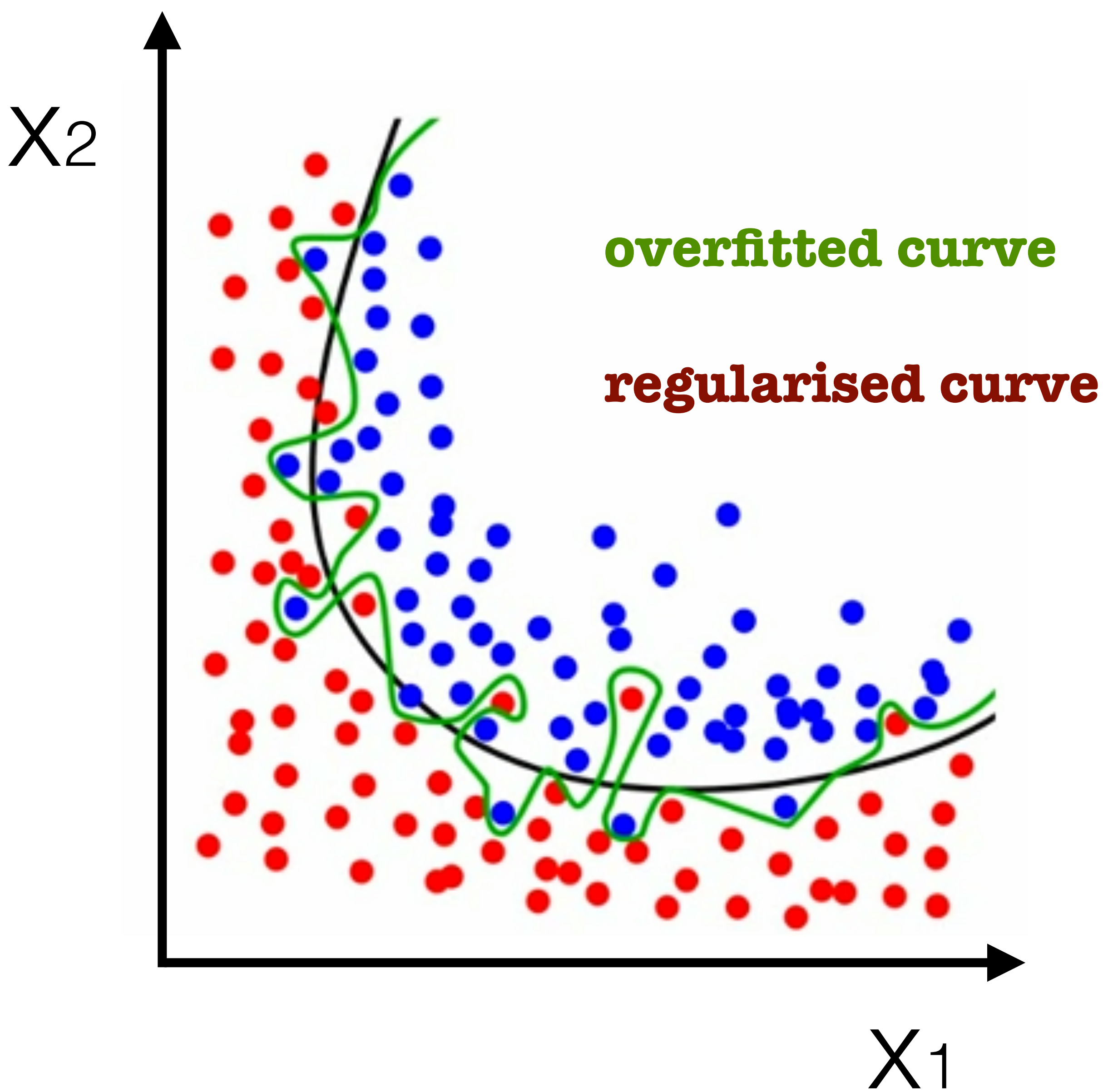
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J$$

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

usually: $\alpha \in [10^{-3}, 1]$

$$\text{scaled } x_j^{(i)} = \frac{x_j^{(i)} - \text{averaged } x_j^{(i)} \text{ in the training sample}}{\text{range of } x_j^{(i)} \text{ in the training sample}}$$

Supervised classification



few objects in training set
+
lots of features with higher degrees

Overfitting

when each feature contributes slightly in classification:

Regularisation

$$J_{reg} = J + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

Penalty term
(Rigid regression)

Confusion Matrix

total = TN + TP + FP + FN

Accuracy (score) = (TN + TP) / total

Error Rate = 1 - Accuracy

Precision = TP / (TP + FP)

TN : True Negative

TP : True Positive

FP : False Positive

FN : False Negative

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Training set selection