Introduction to Machine Learning

Outline

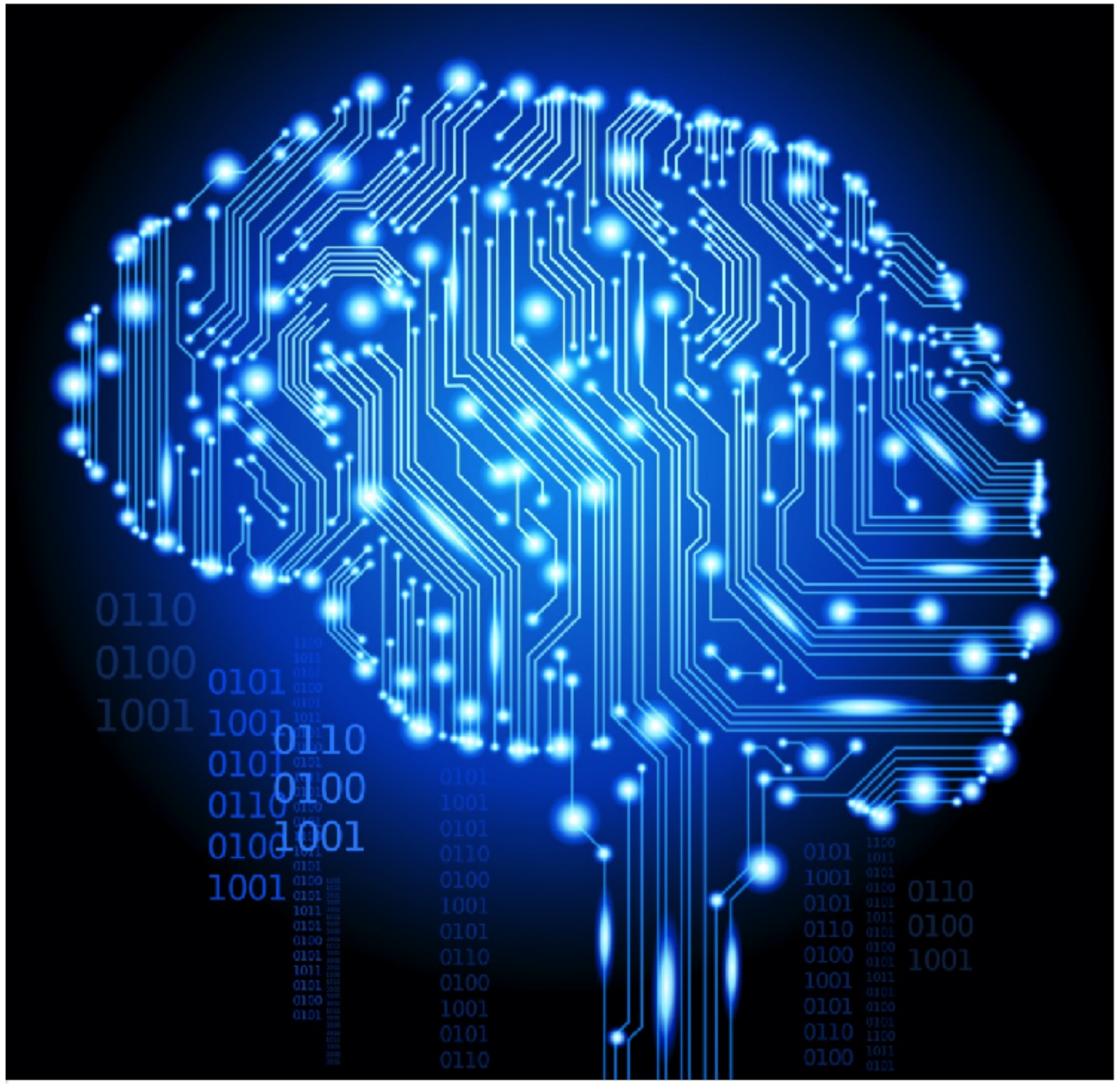
- Overview
- Supervised Machine Learning
- Regression versus Classification
- Regularization
- Unsupervised Machine Learning
- Dimensionality Reduction versus Clustering

Overview

- According to Arthur Samuel (1959), Machine Learning (ML) gives computers the ability to learn without being explicitly programmed

- ML is useful to solve a problem when it isn't possible to develop a deterministic polynomial code

- ML makes predictions or decision from data analysis



Types of Machine Learning

	Supervised Machine Learning	Unsupervised Machine Learning
Discrete Data	Classification	Clustering
Continuous Data	Regression	Dimensionality Reduction

Types of Machine Learning

	Supervised Machine Learning	Unsupervised Machine Learning
Discrete Data	Classification	Clustering
Continuous Data	Regression	Dimensionality Reduction

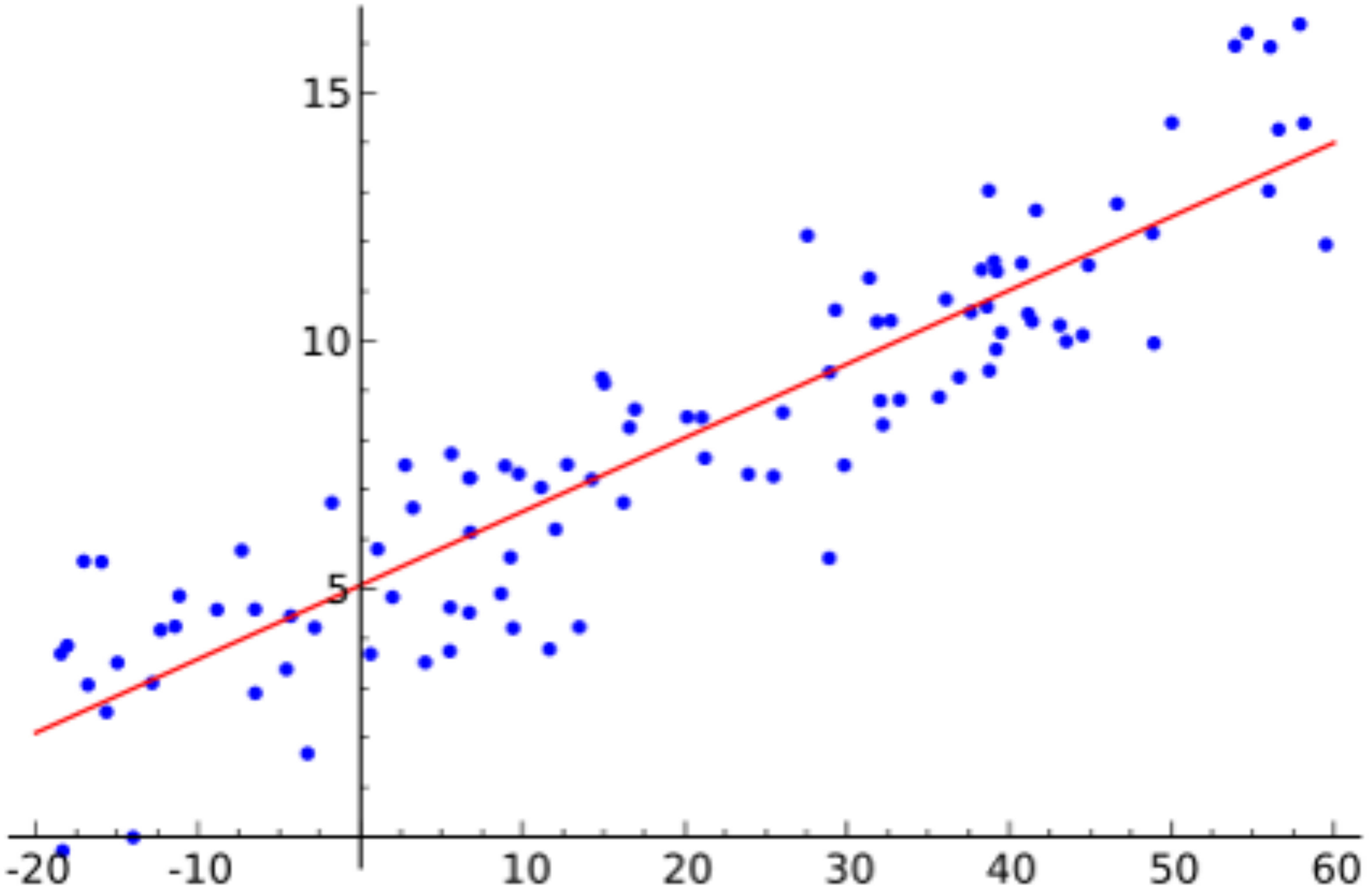
Supervised Machine Learning

Supervised Machine Learning

- In a supervised learning problem, we have a dataset with real data from the problem at hand
- Based on the available information, we can establish a hypothesis for future predict output data
- Continuous outputs: **regression problem**
- Discrete outputs: **classification problem**

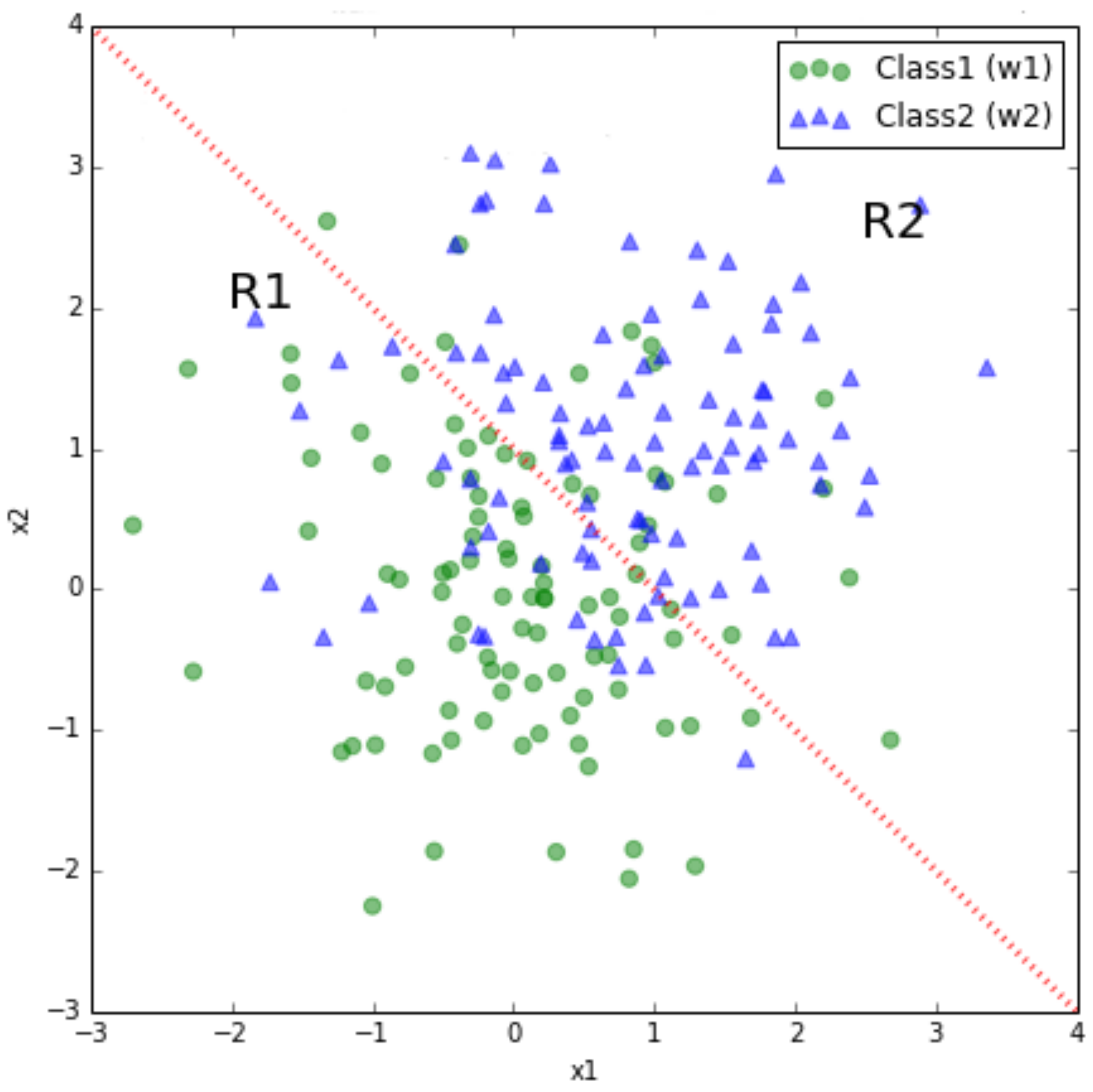
Supervised Machine Learning

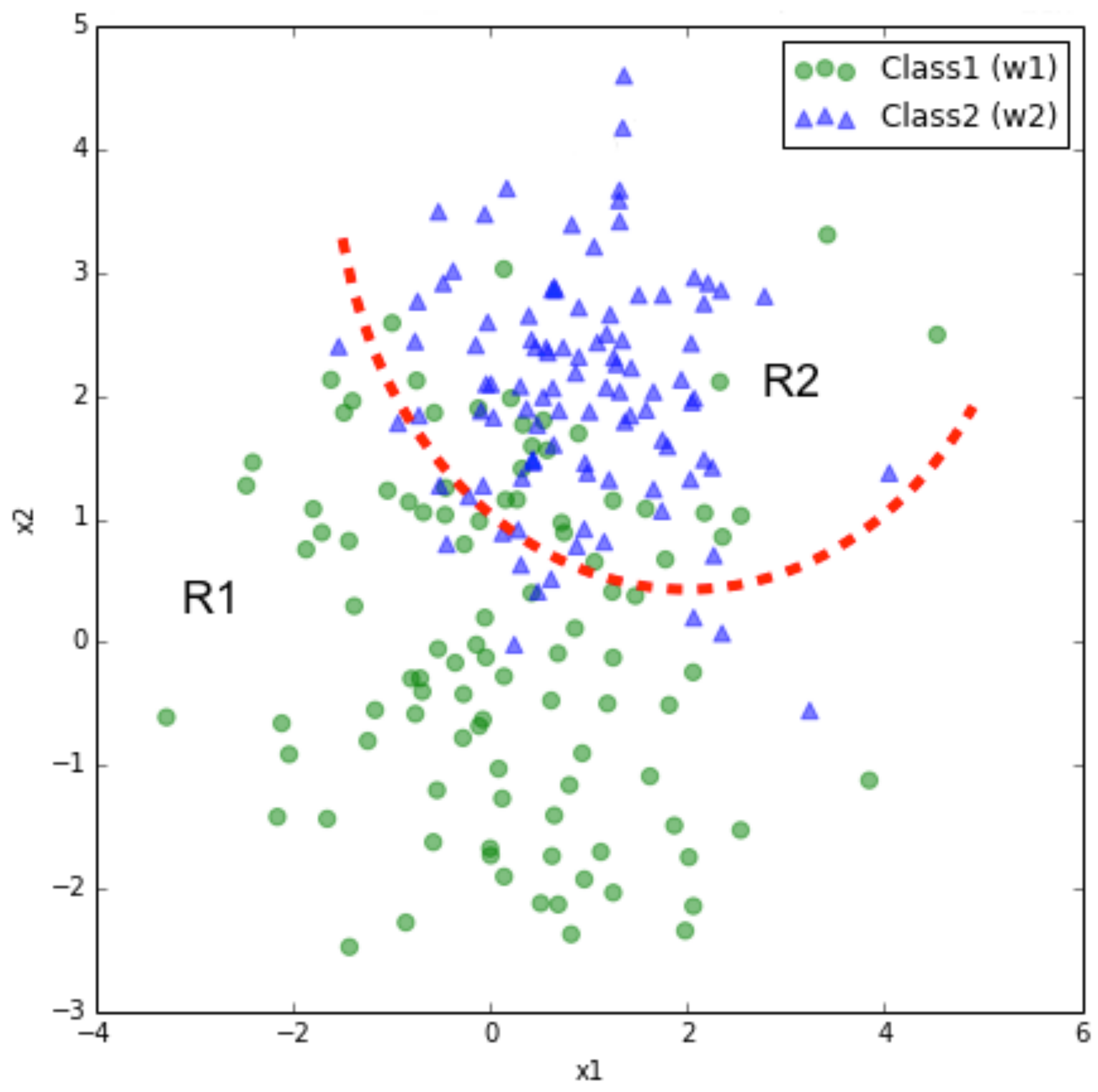
- **Regression:** are used to predict parameters, e.g. weight versus height



Supervised Machine Learning

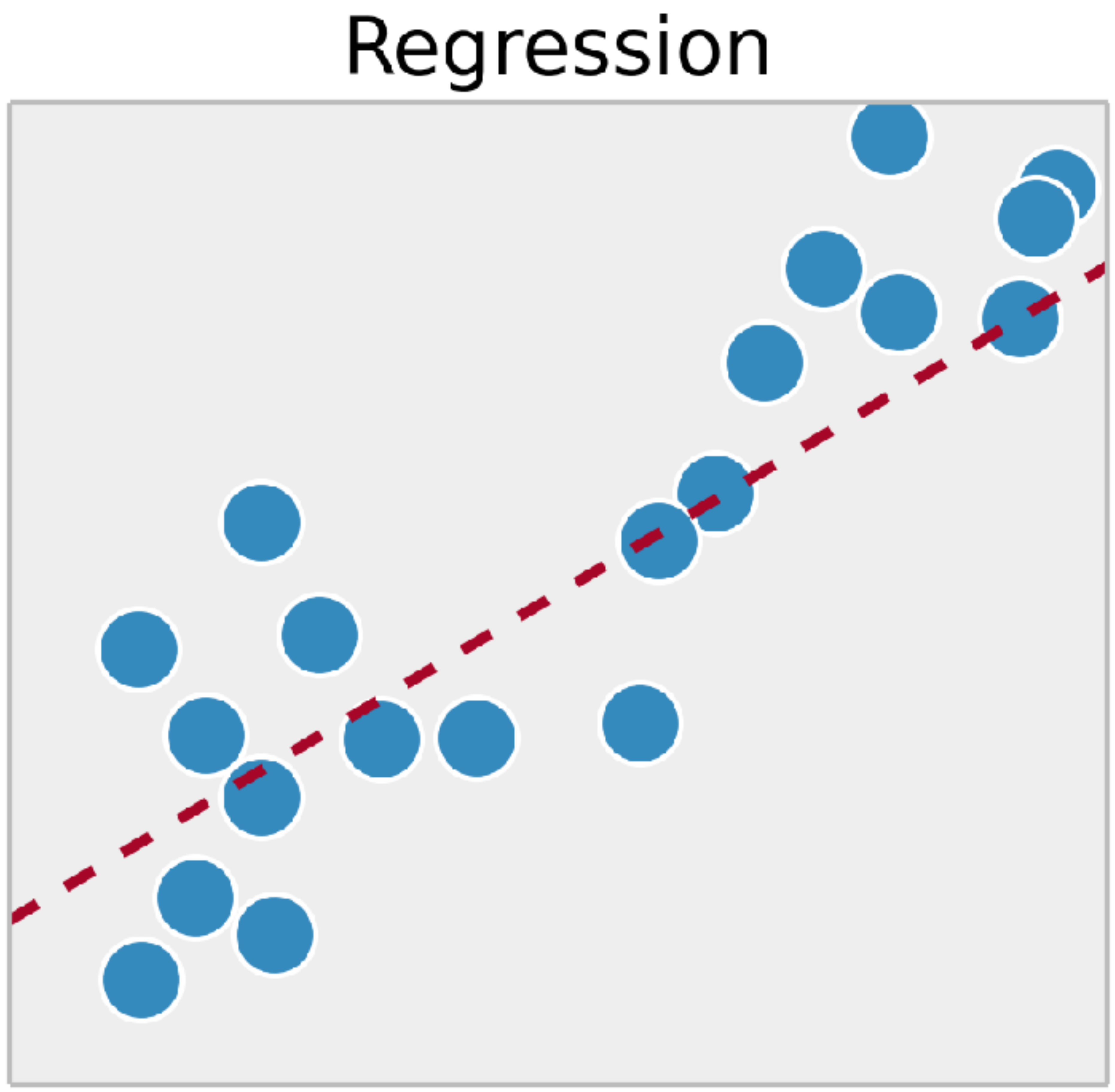
- **Classification**: are used to predict classes, e.g. disease pre-diagnosis





Supervised Machine Learning

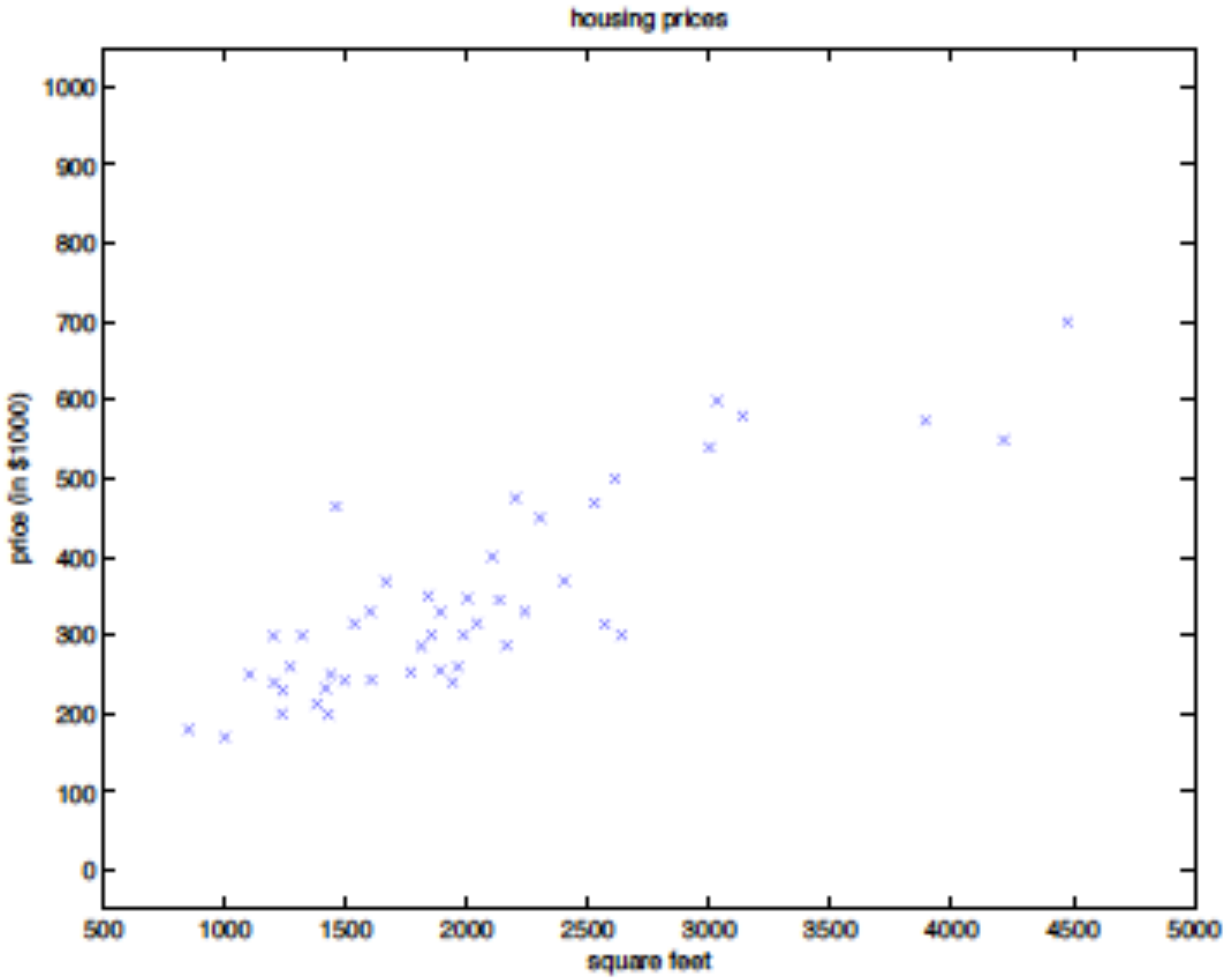
Regression



Example of Supervised ML Problem

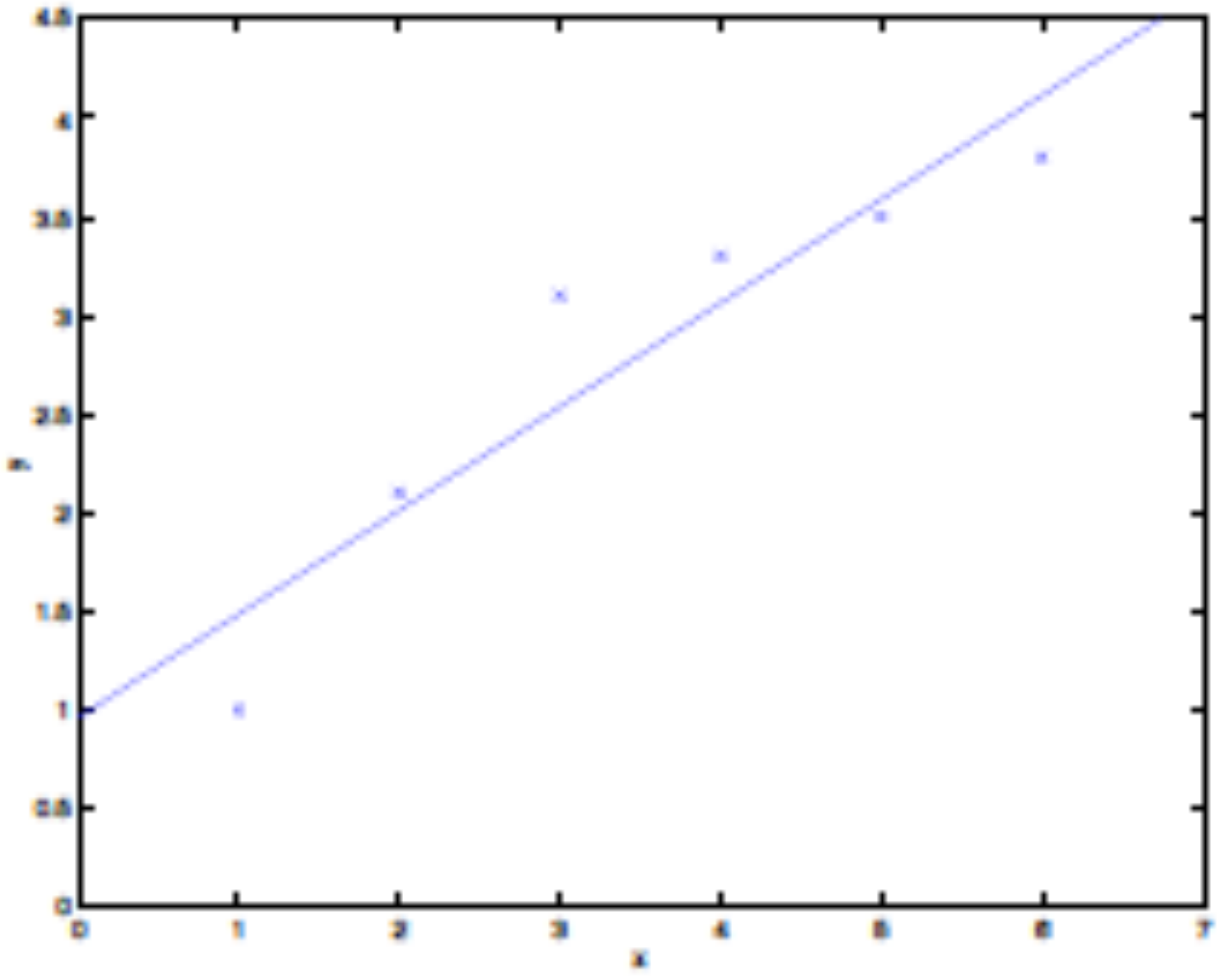
- This dataset shows the prices of houses in Portland/USA, as a function of the size of their living areas
- Is it possible to predict the prices of other houses based on this information?

Living area ($feet^2$)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

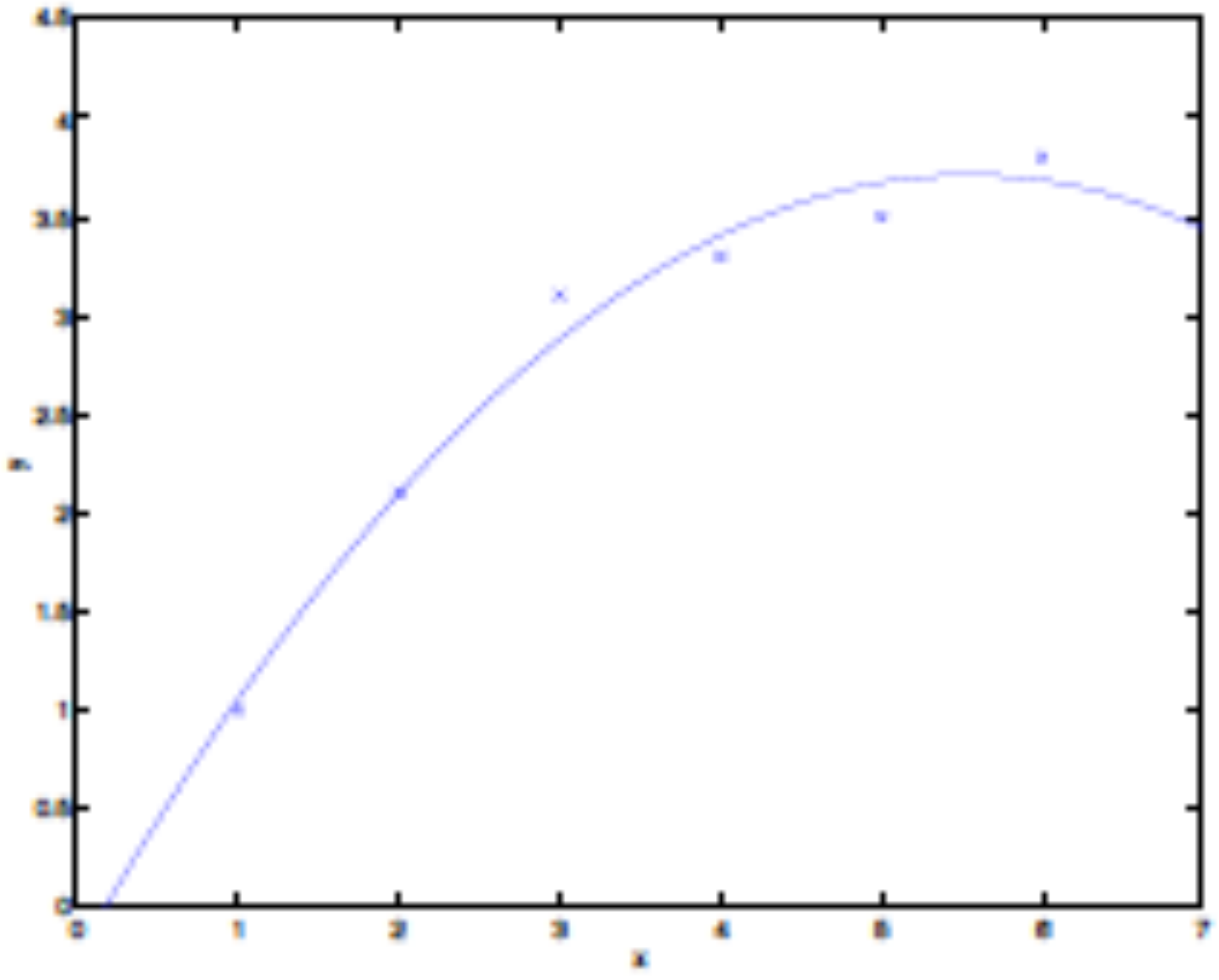


Fitting the Input Data

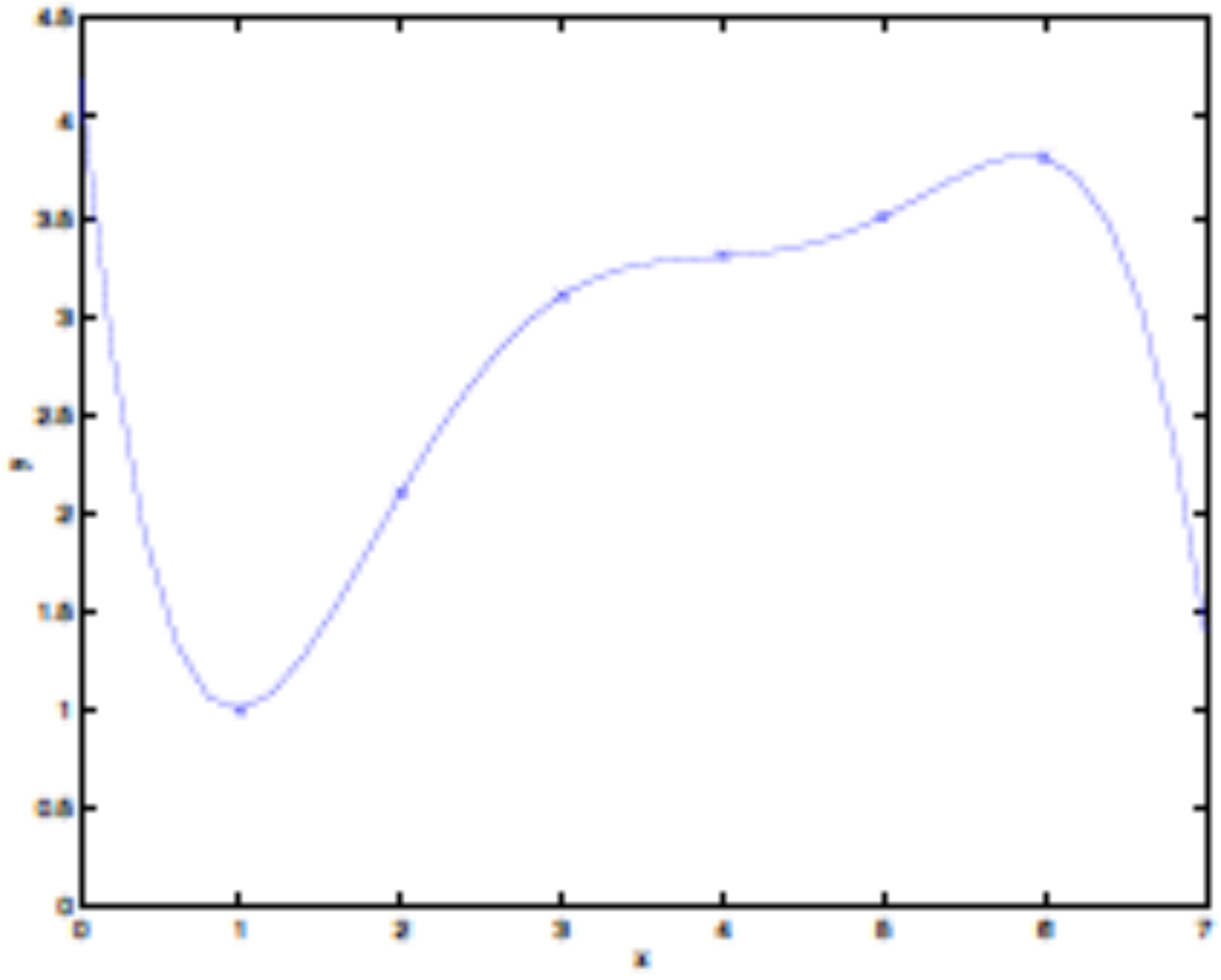
Linear Model



Second Order Polynomial

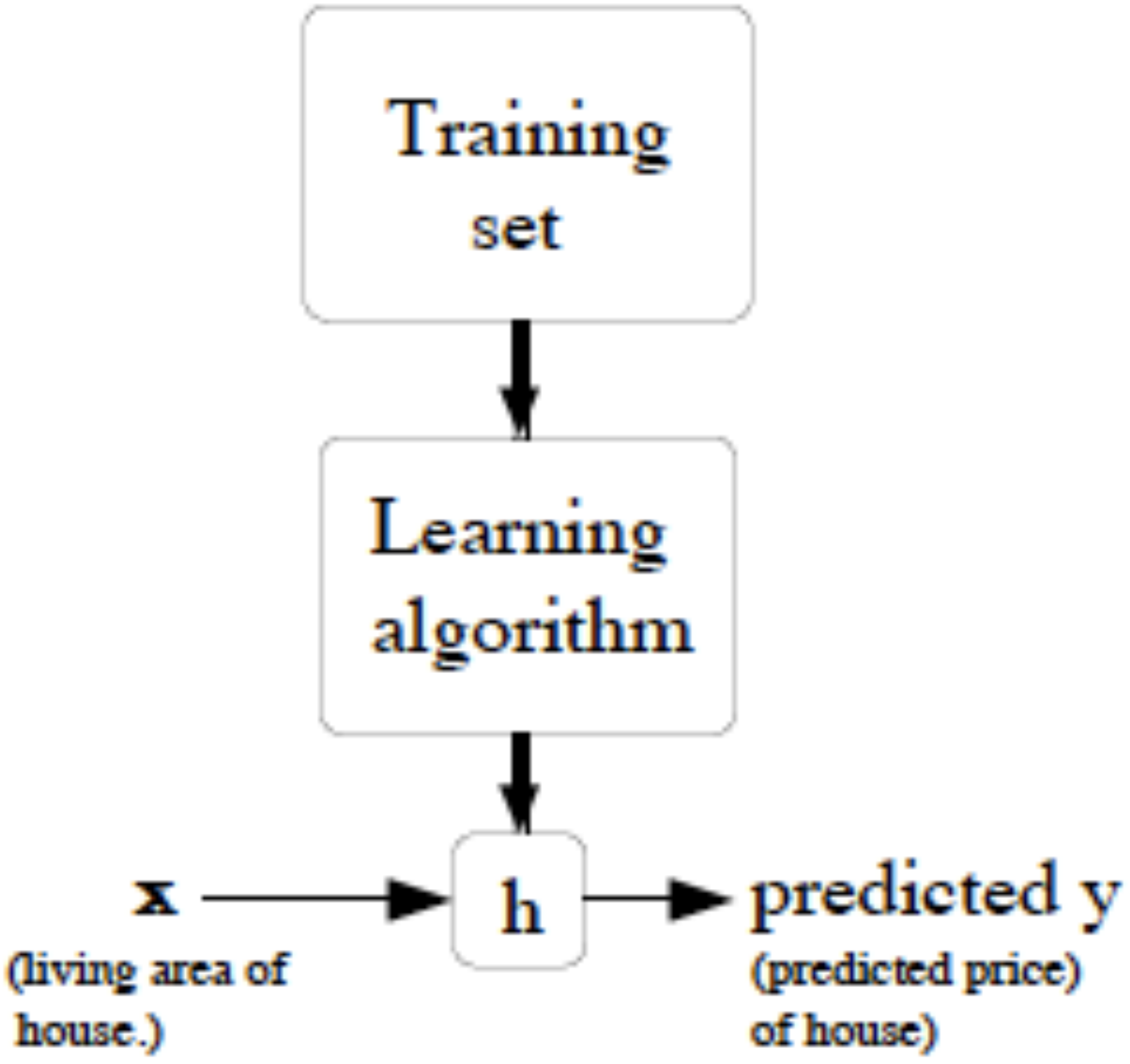


N-Order Polynomial



Supervised Regression ML

- Given a training set, to learn a function h = X → Y so that h(x) is a "good" predictor for the corresponding value of y.



Linear Regression

- Hypothesis

$$h_\theta(x) = \theta_0 + \theta_1 x_1 \qquad h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

- Cost function

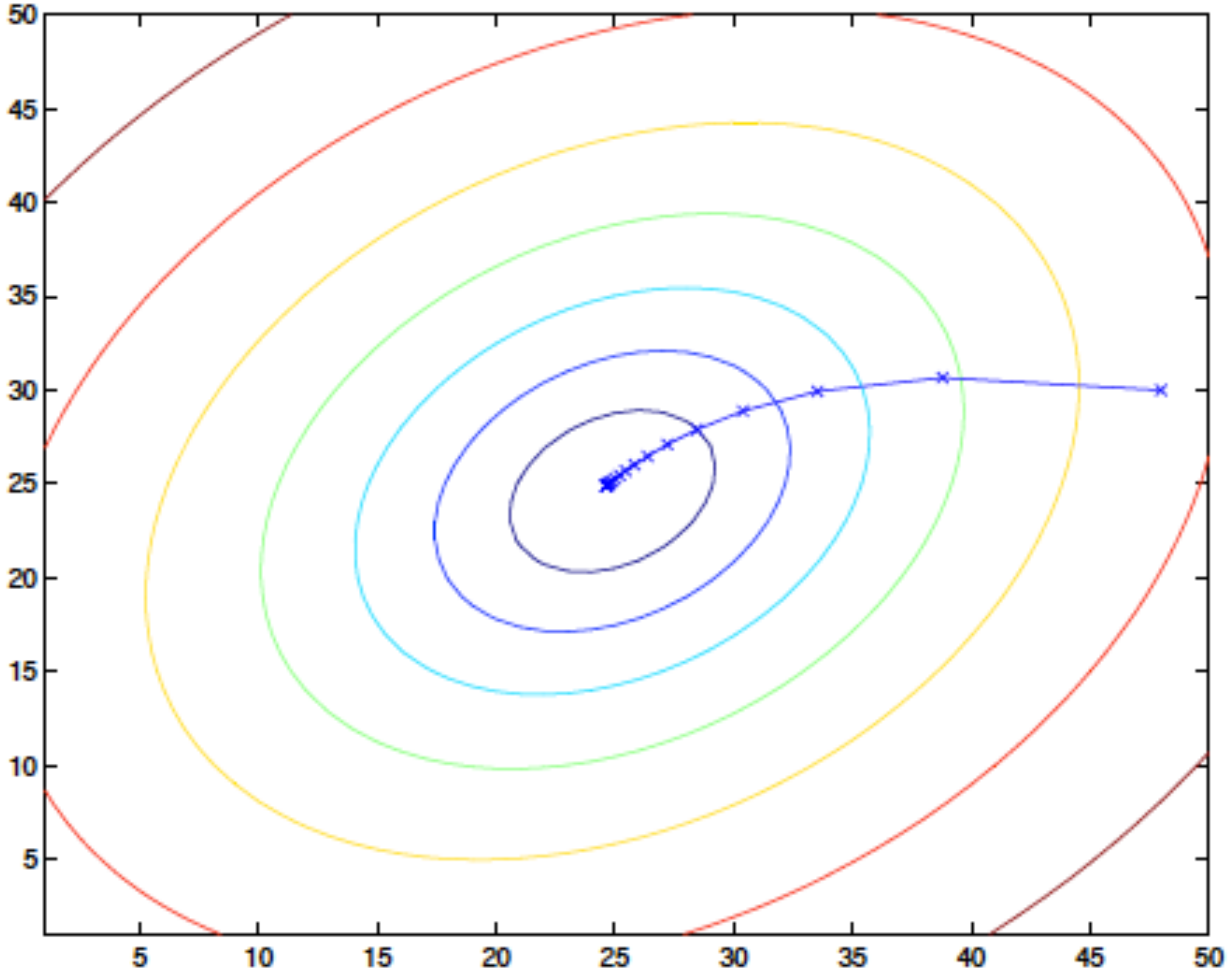
$$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

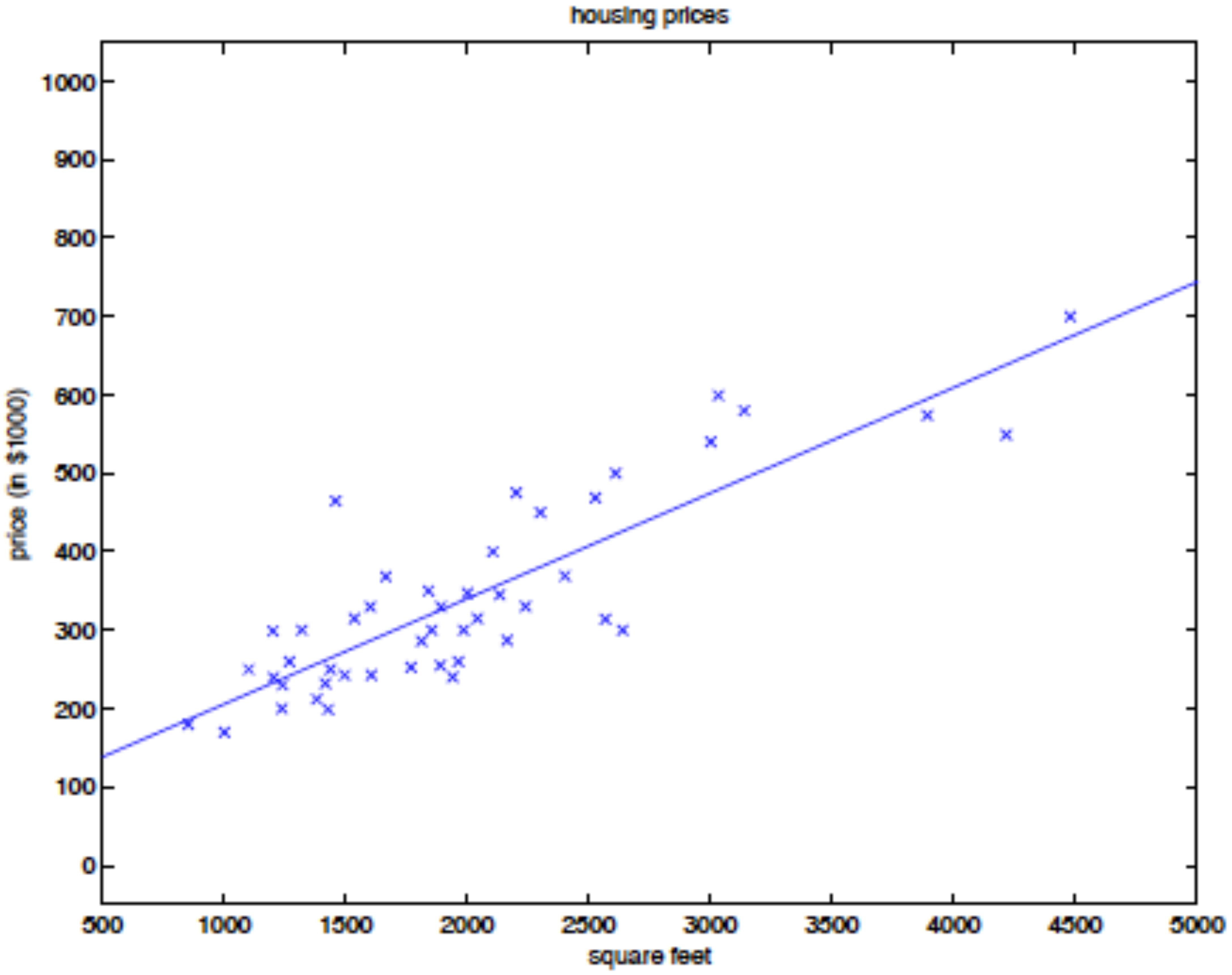
- Goal

$$\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$$

Supervised Regression ML

- Gradient Descent



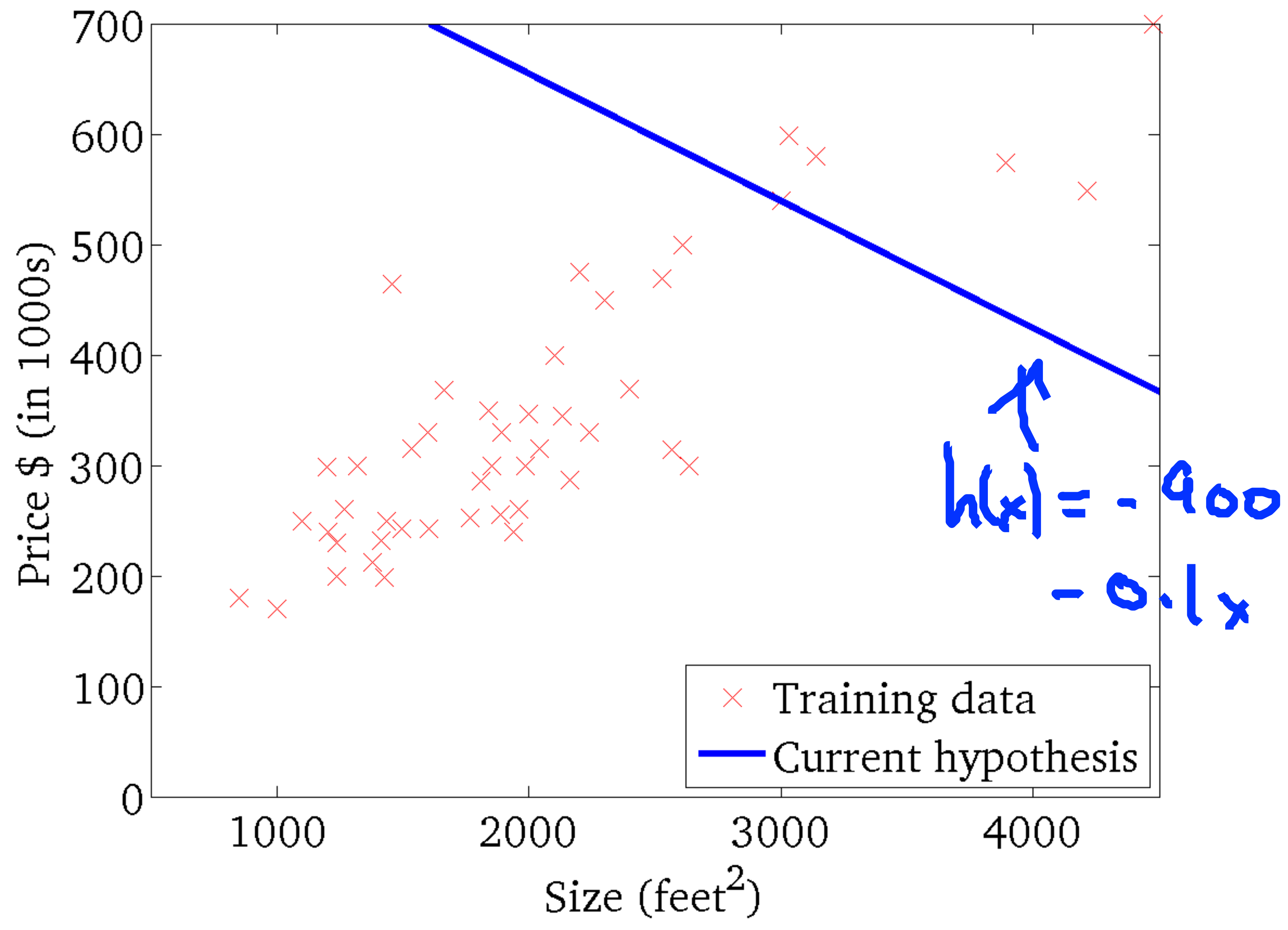


Supervised Regression ML

- Gradient Descent

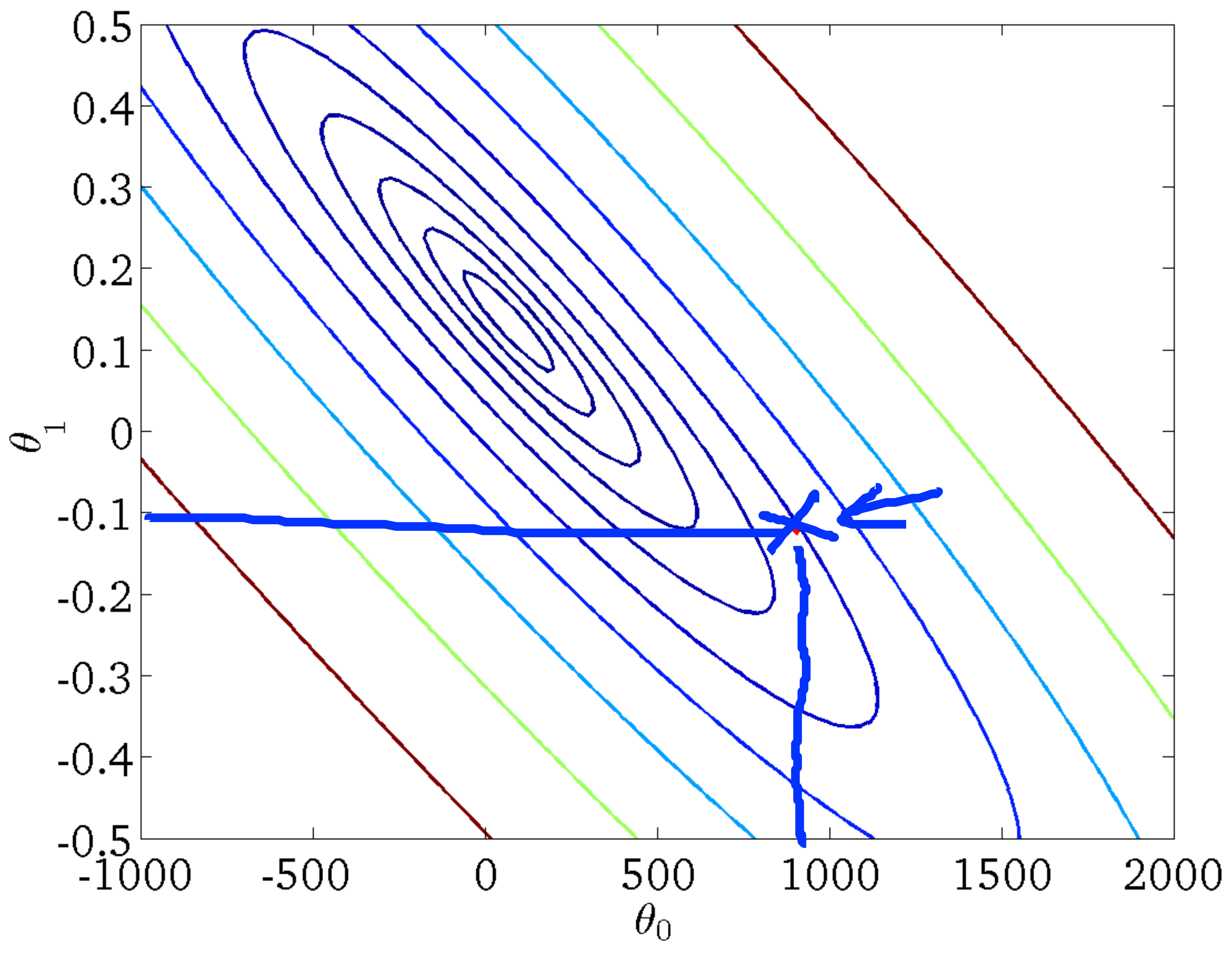
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



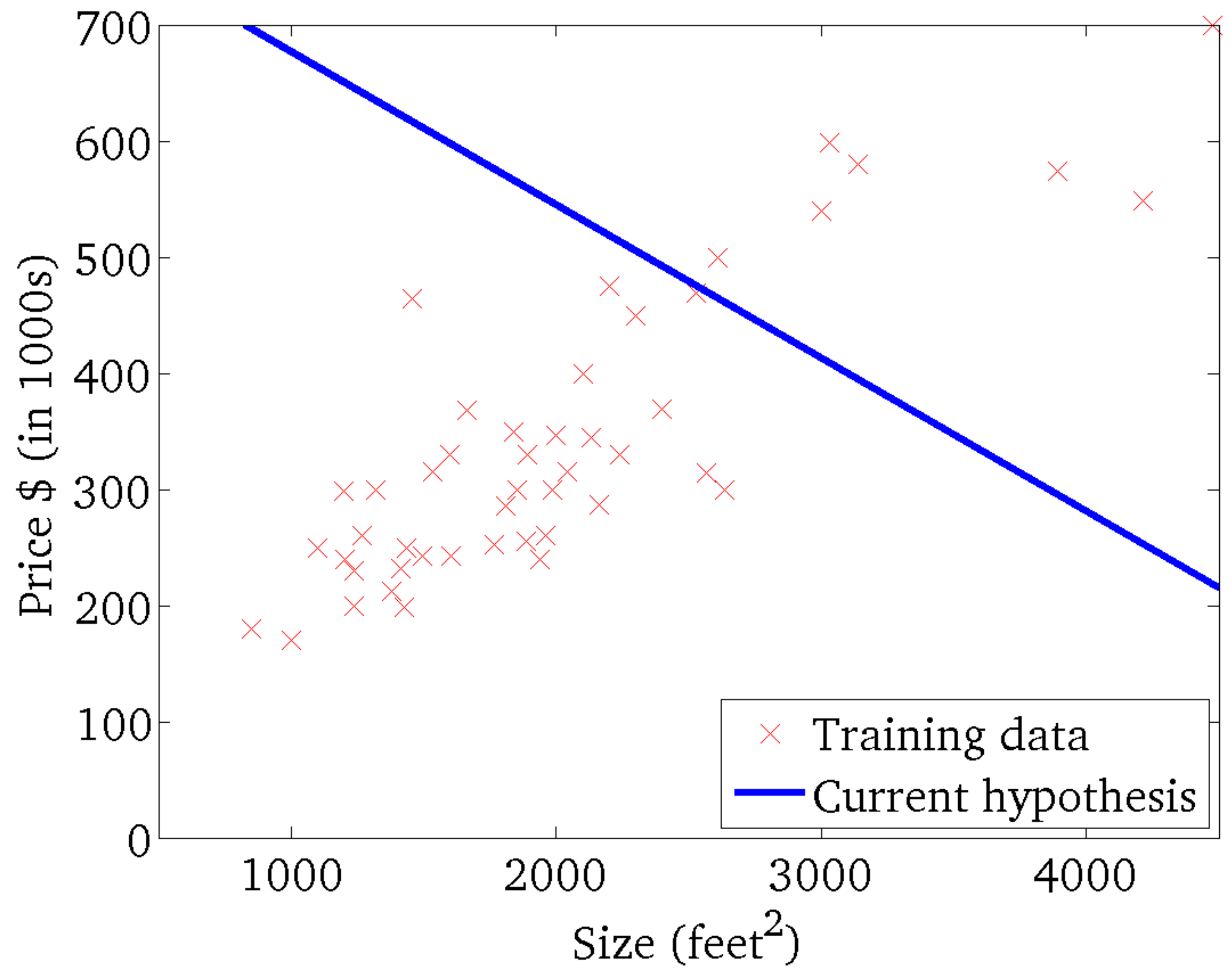
Andrew Ng

Supervised Regression ML

- Gradient Descent

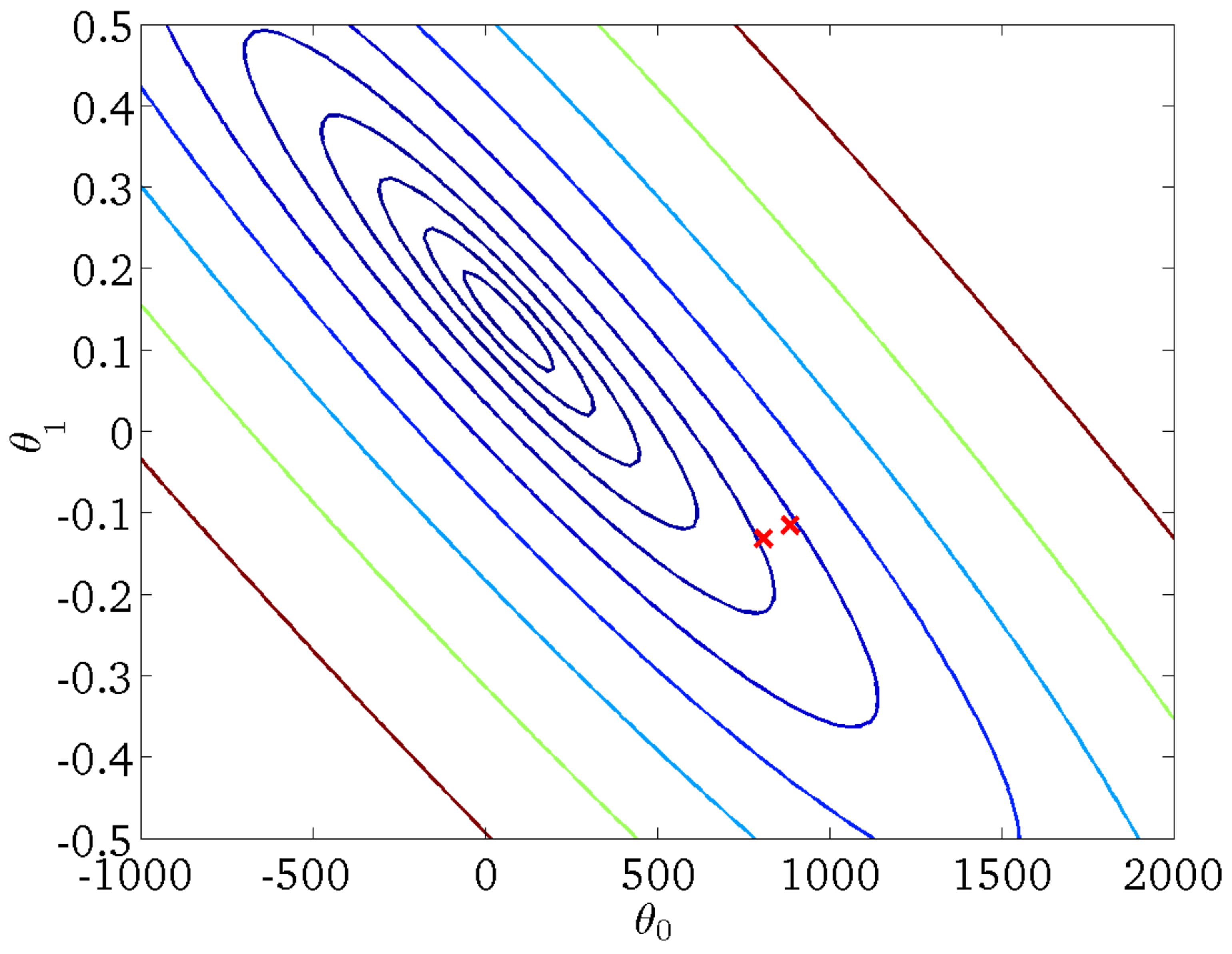
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



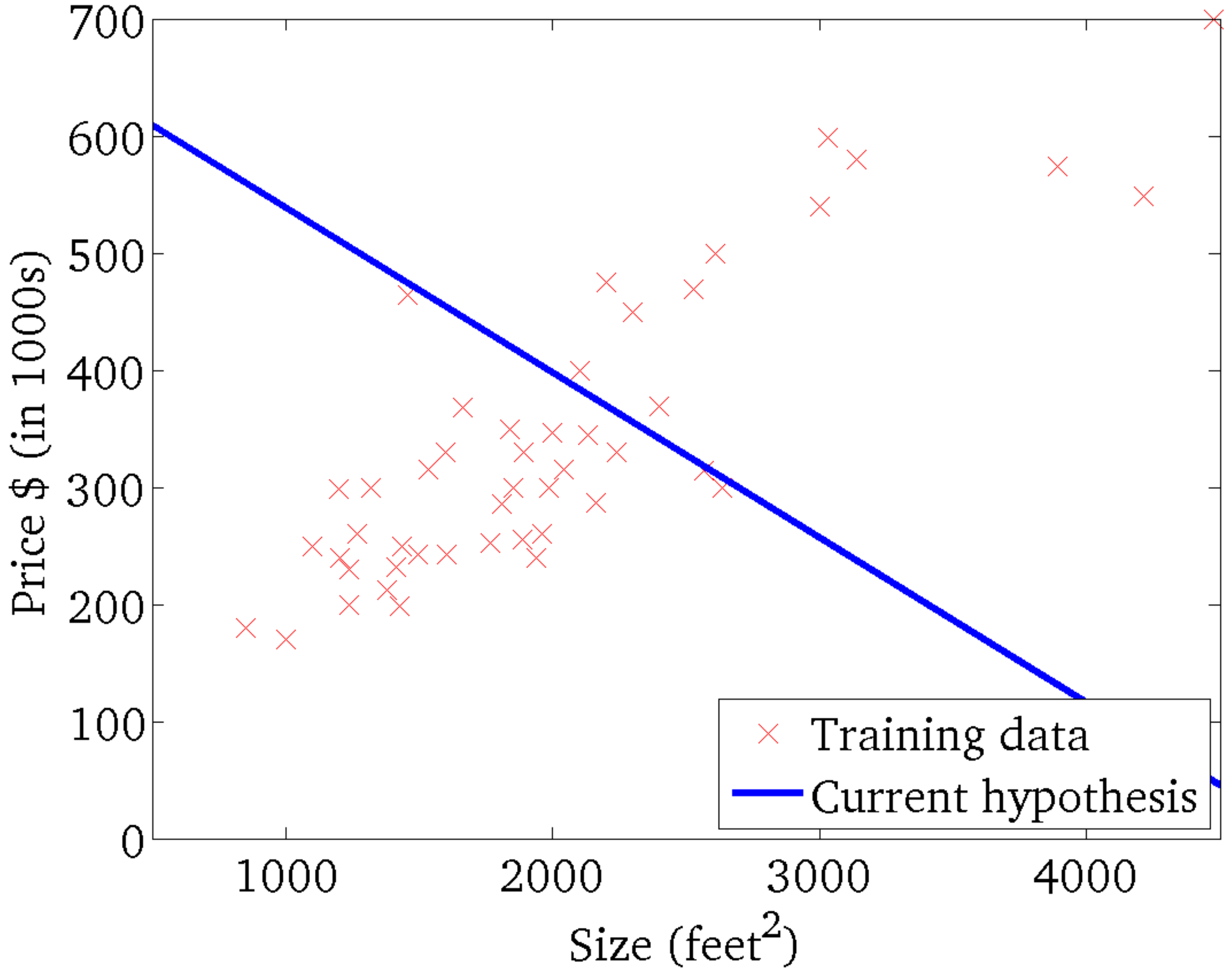
Andrew Ng

Supervised Regression ML

- Gradient Descent

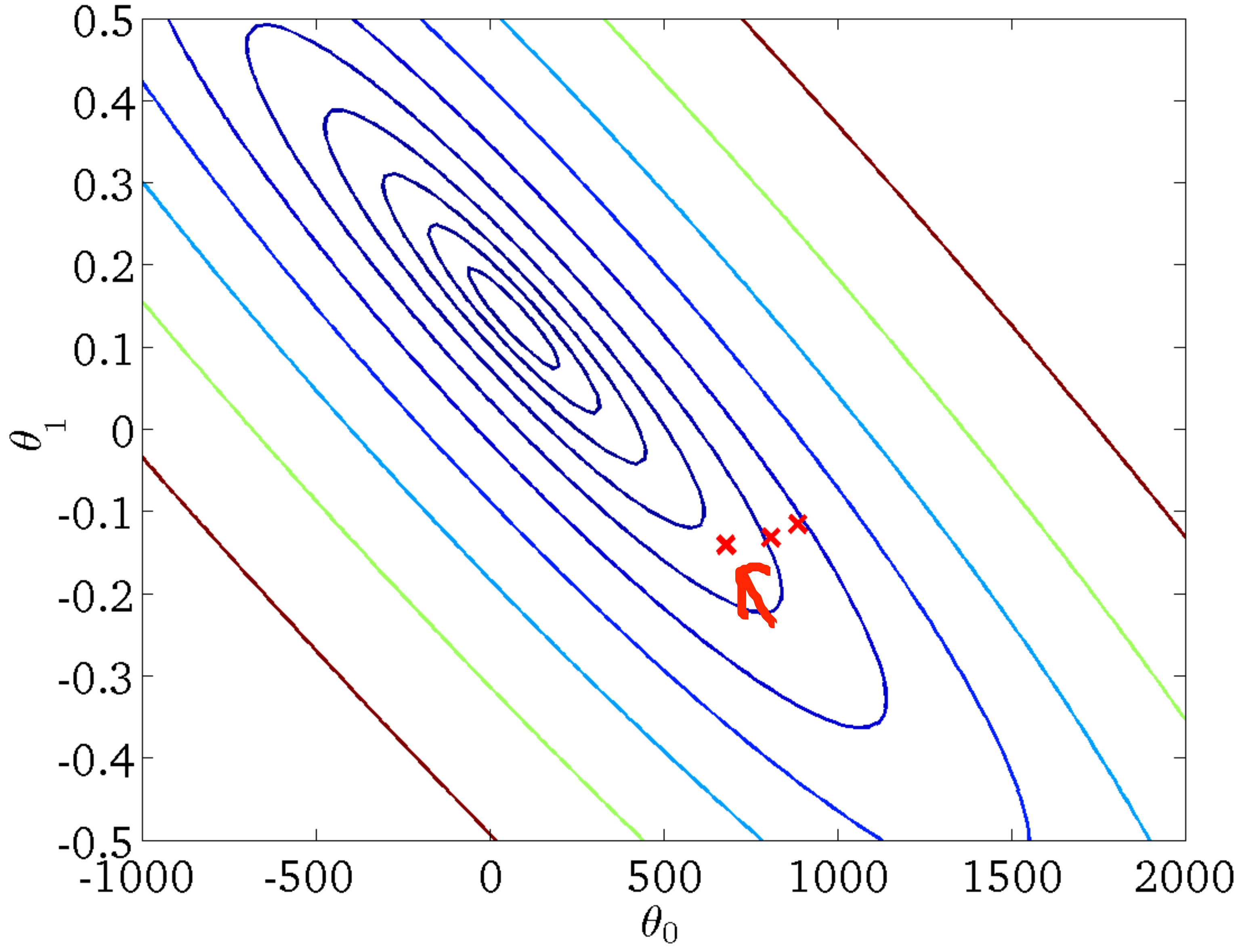
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



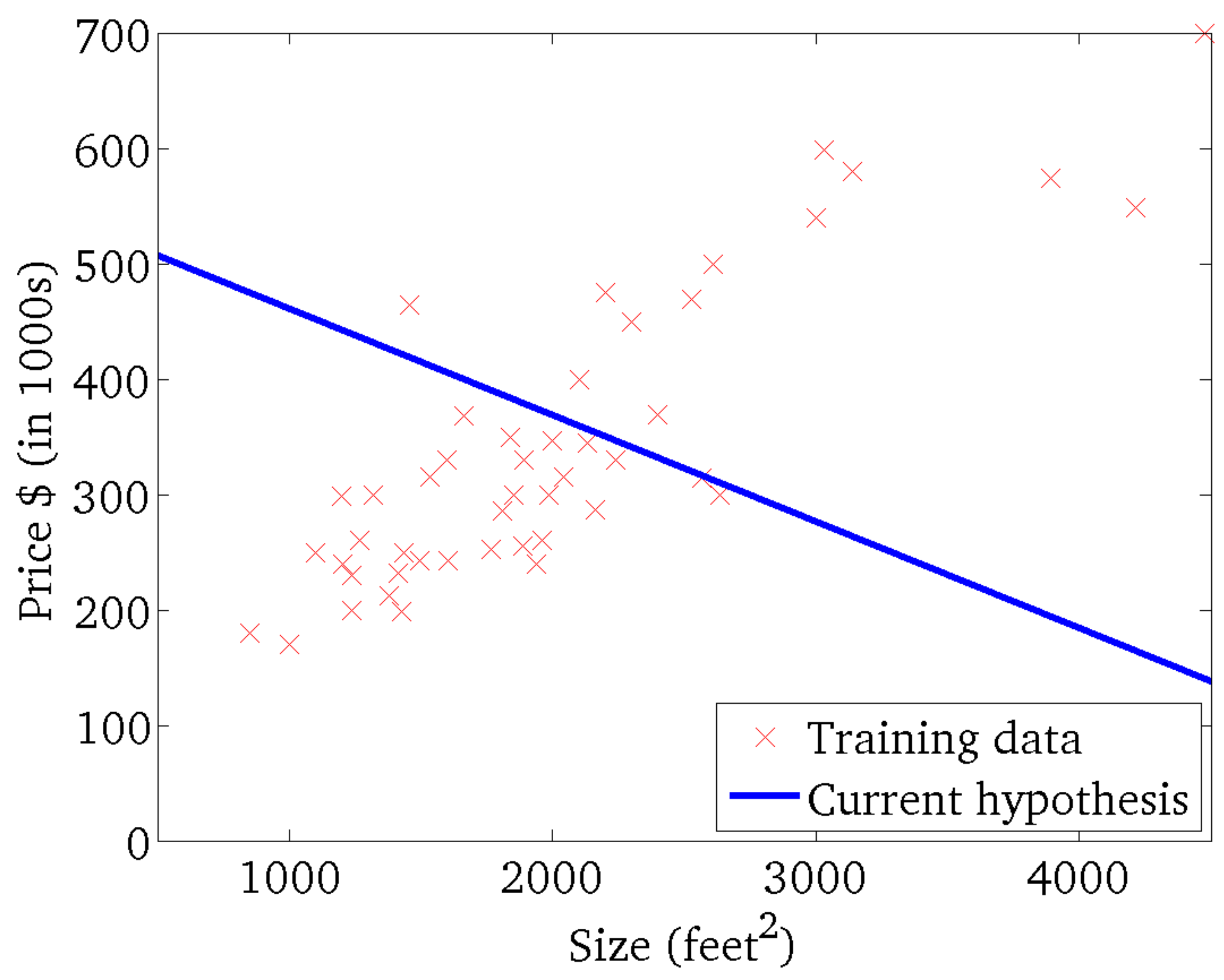
Andrew Ng

Supervised Regression ML

- Gradient Descent

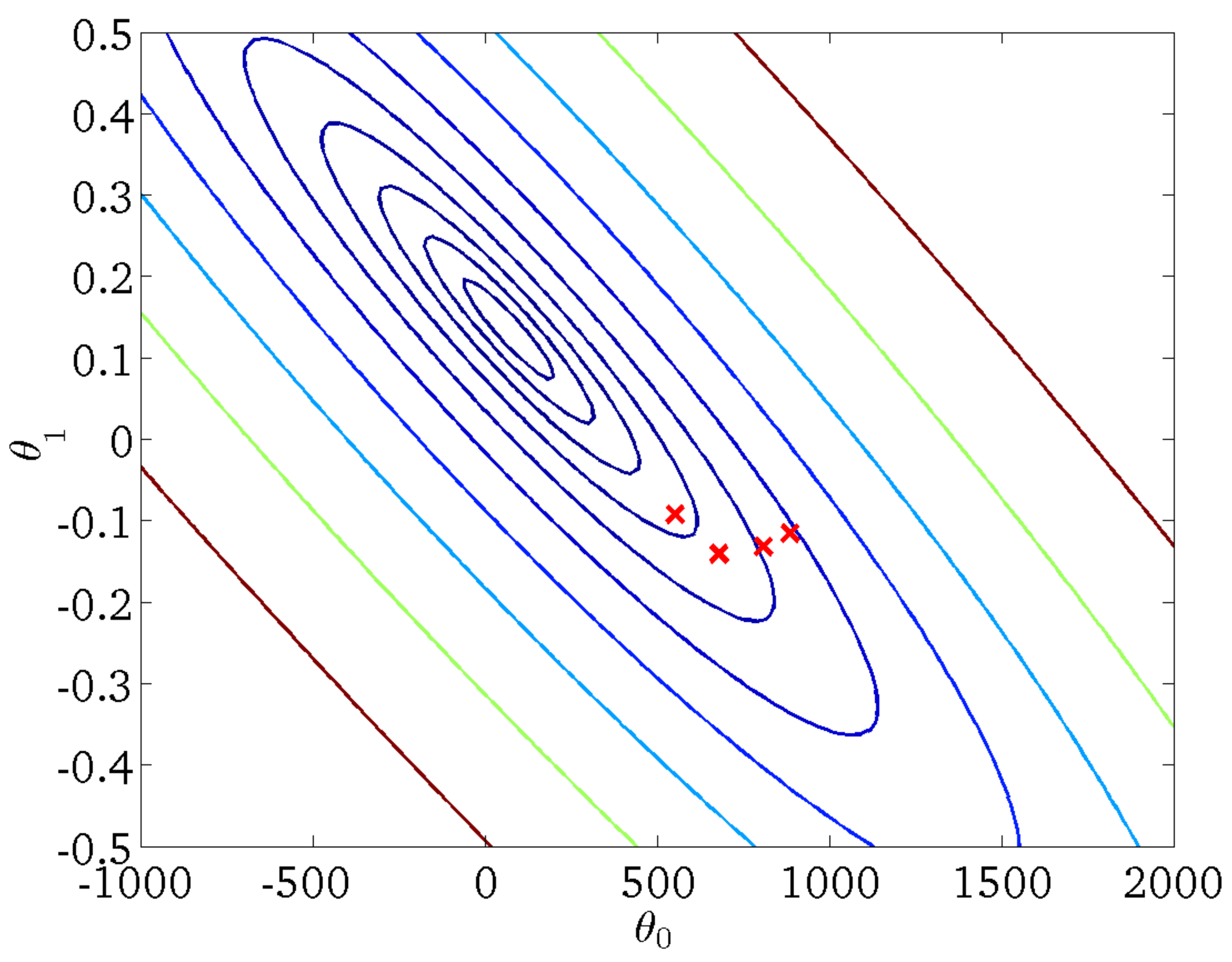
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



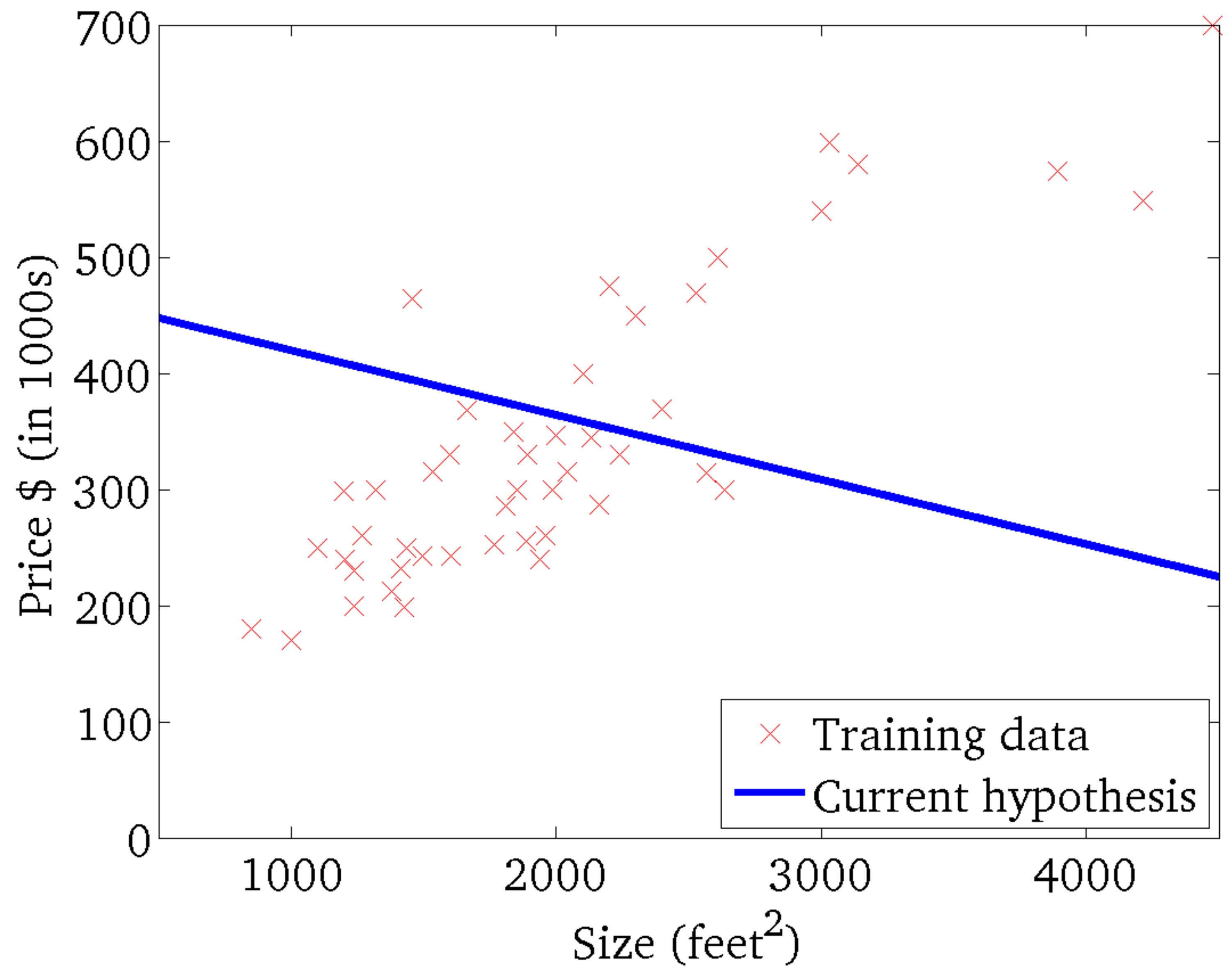
Andrew Ng

Supervised Regression ML

- Gradient Descent

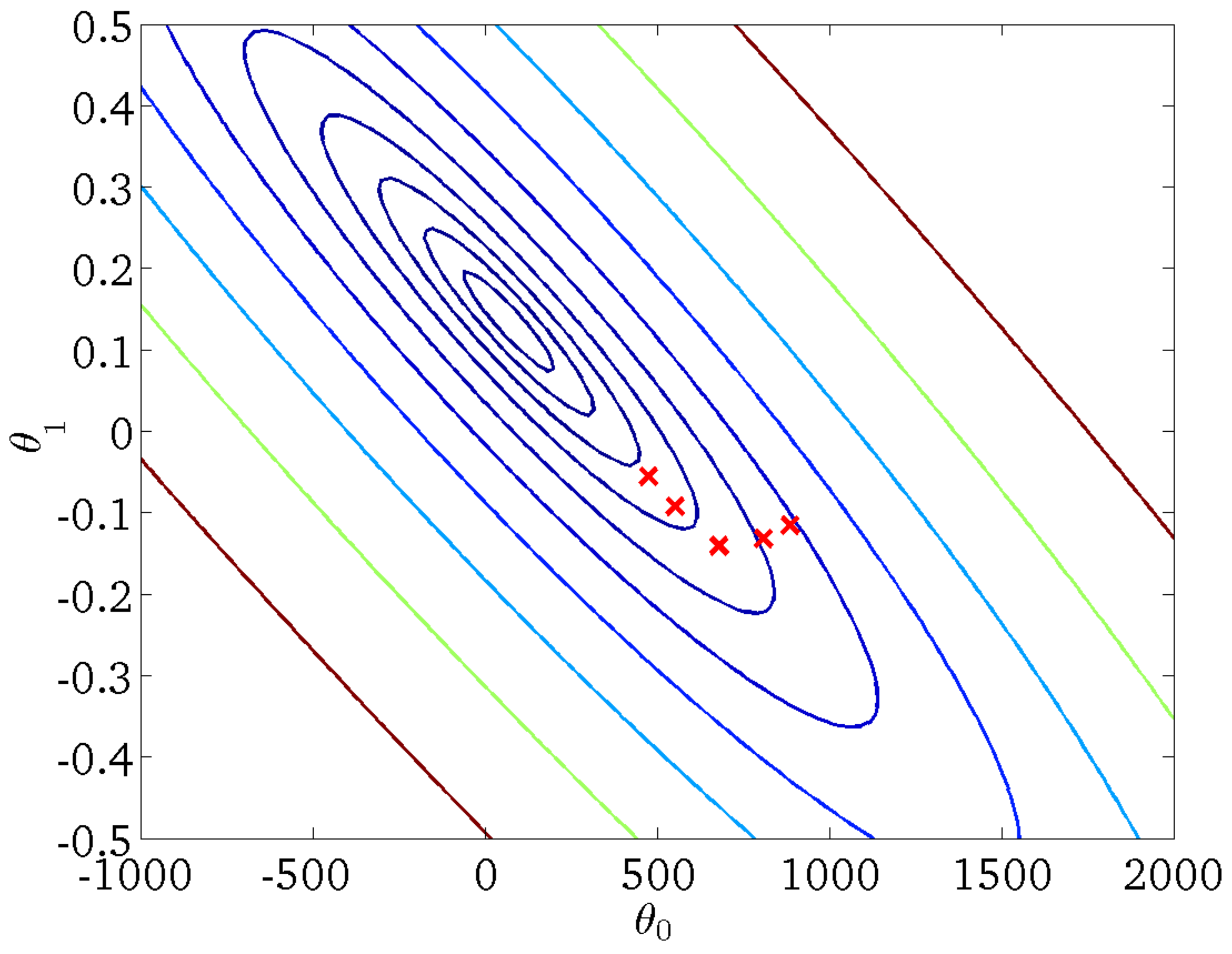
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



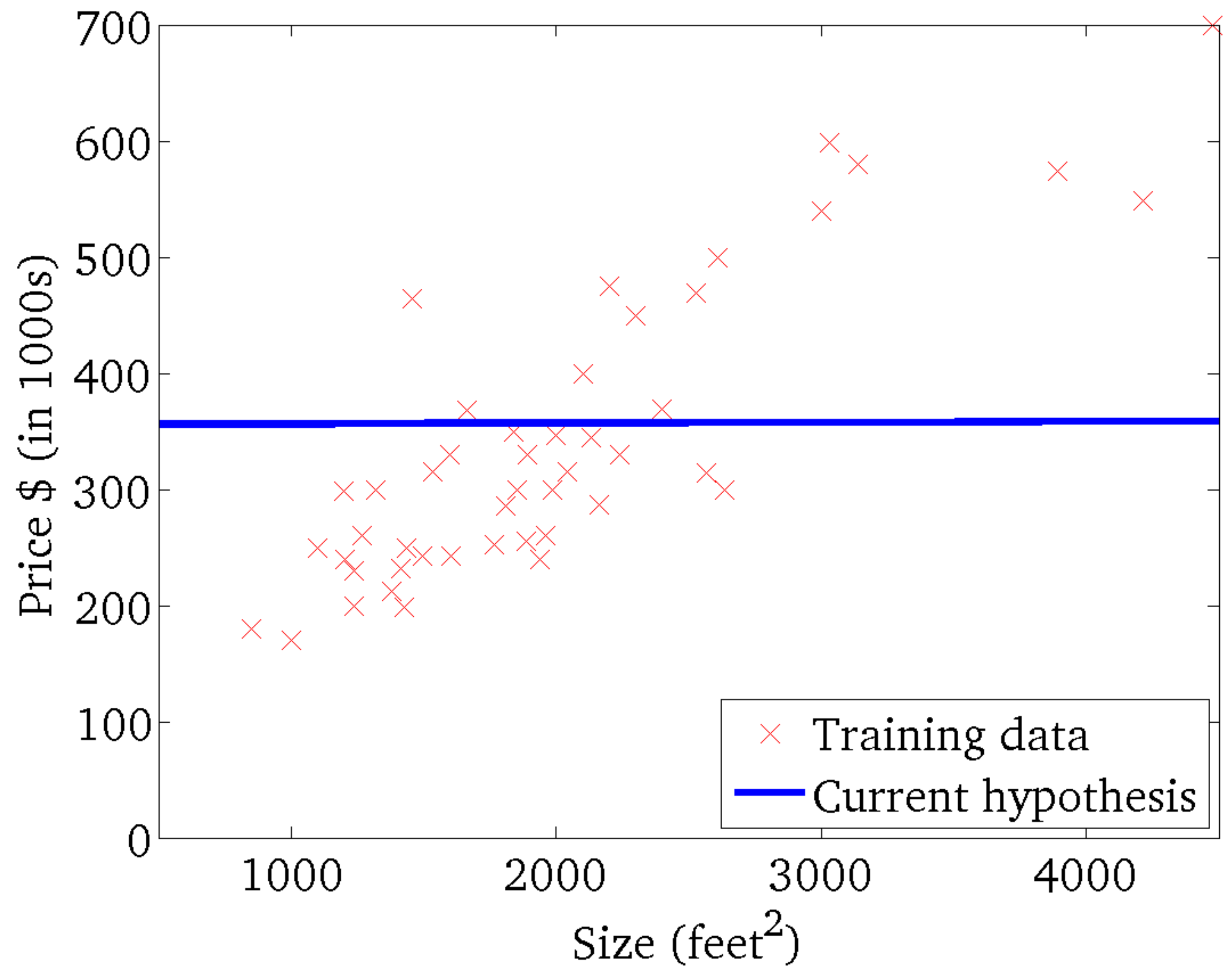
Andrew Ng

Supervised Regression ML

- Gradient Descent

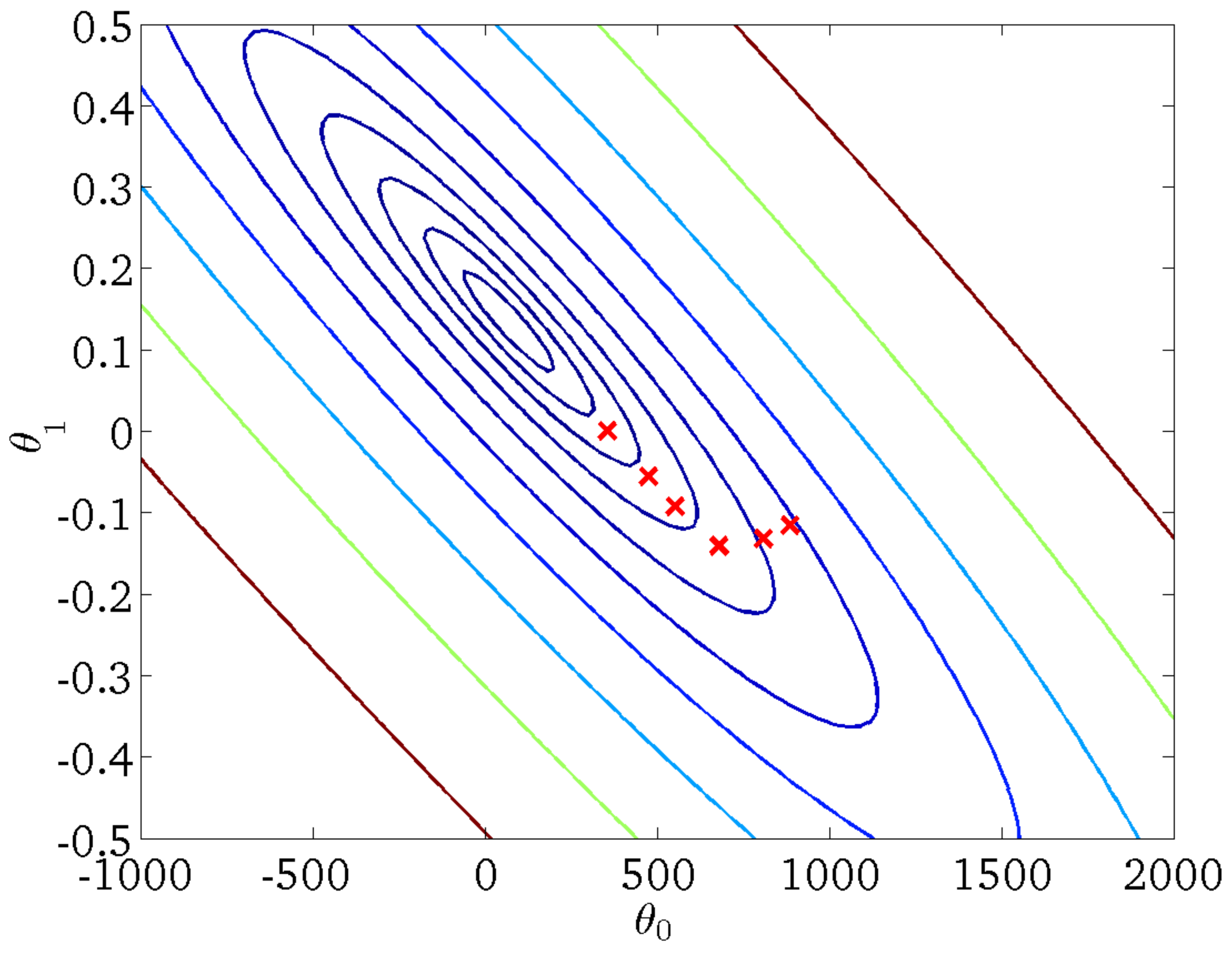
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



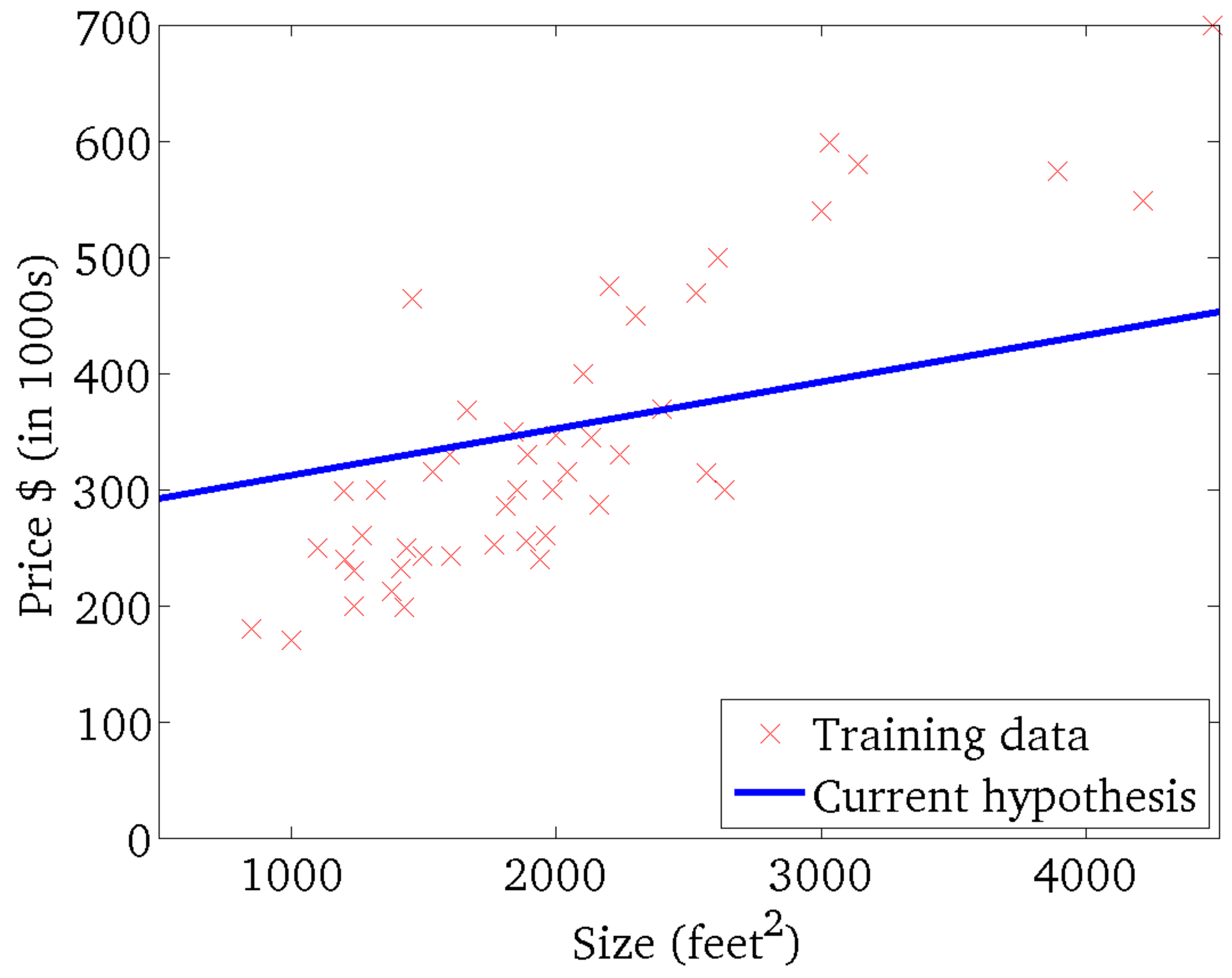
Andrew Ng

Supervised Regression ML

- Gradient Descent

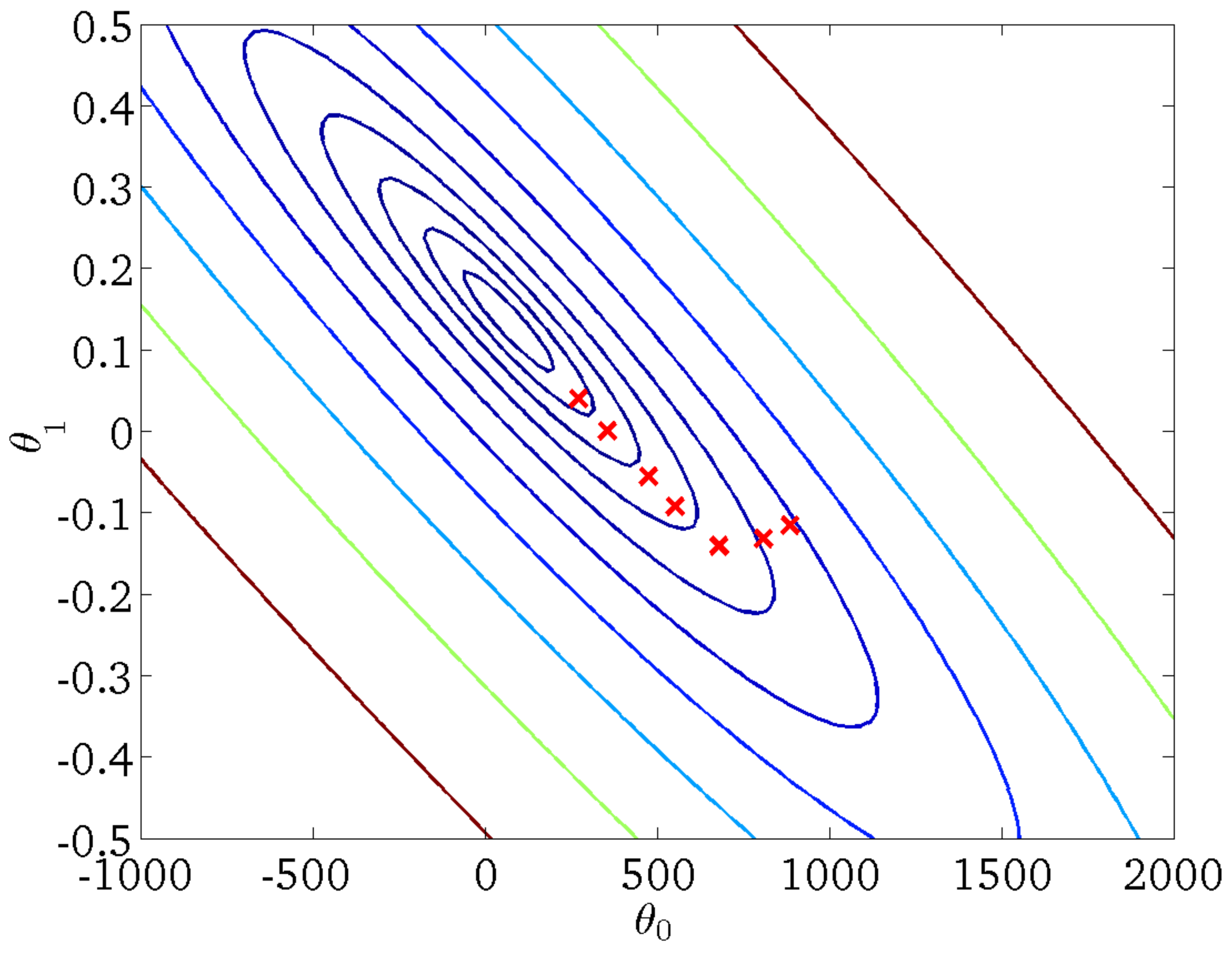
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



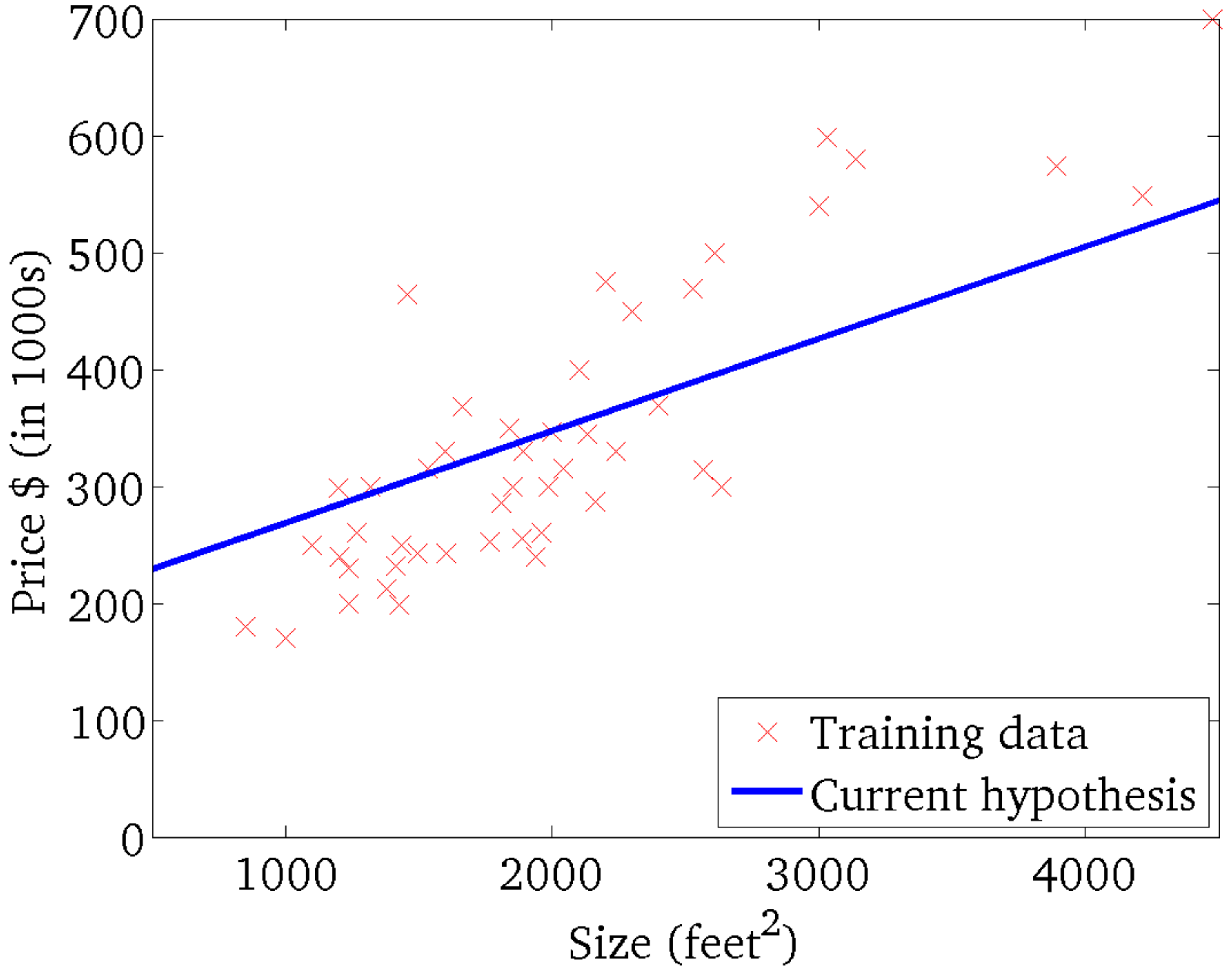
Andrew Ng

Supervised Regression ML

- Gradient Descent

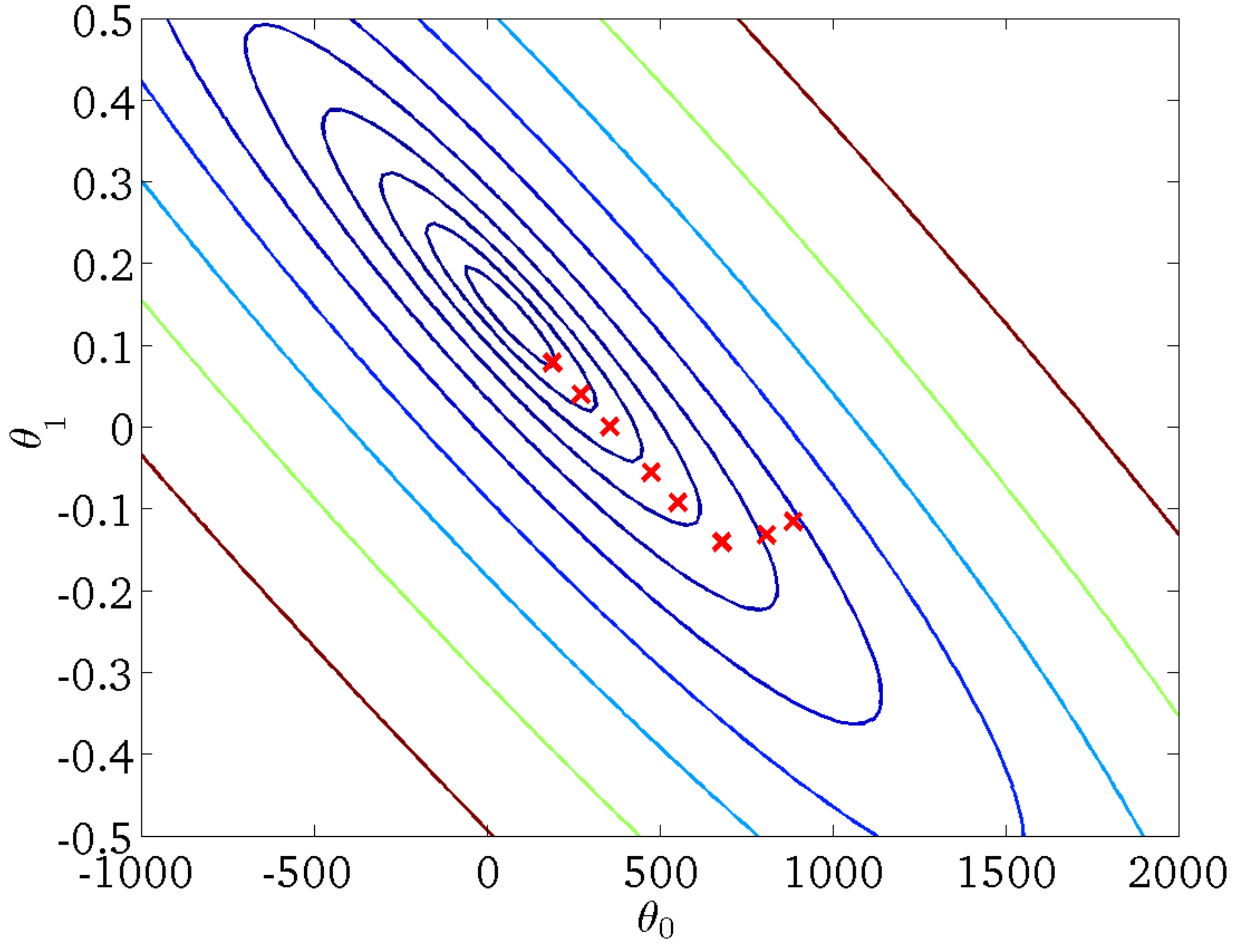
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



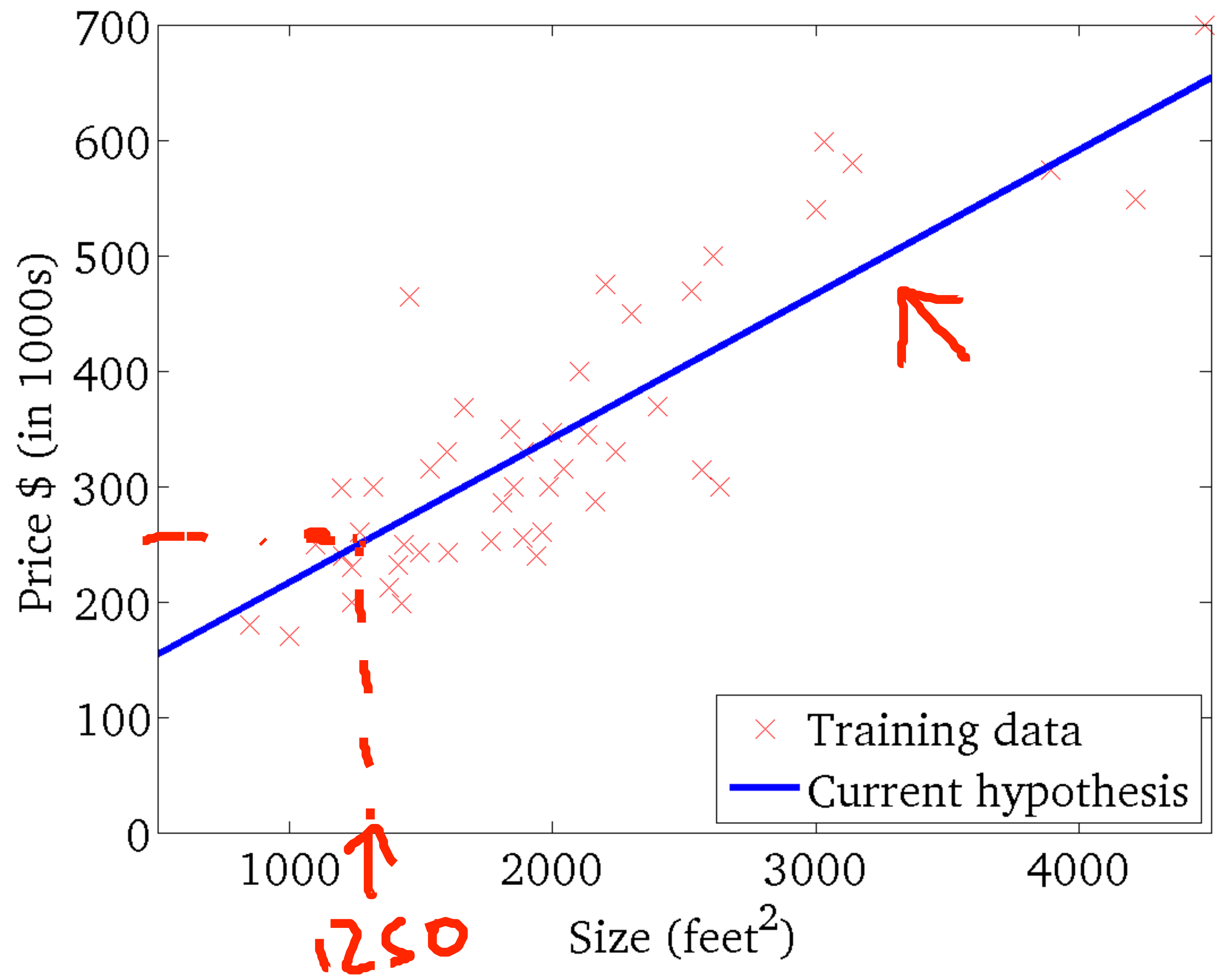
Andrew Ng

Supervised Regression ML

- Gradient Descent

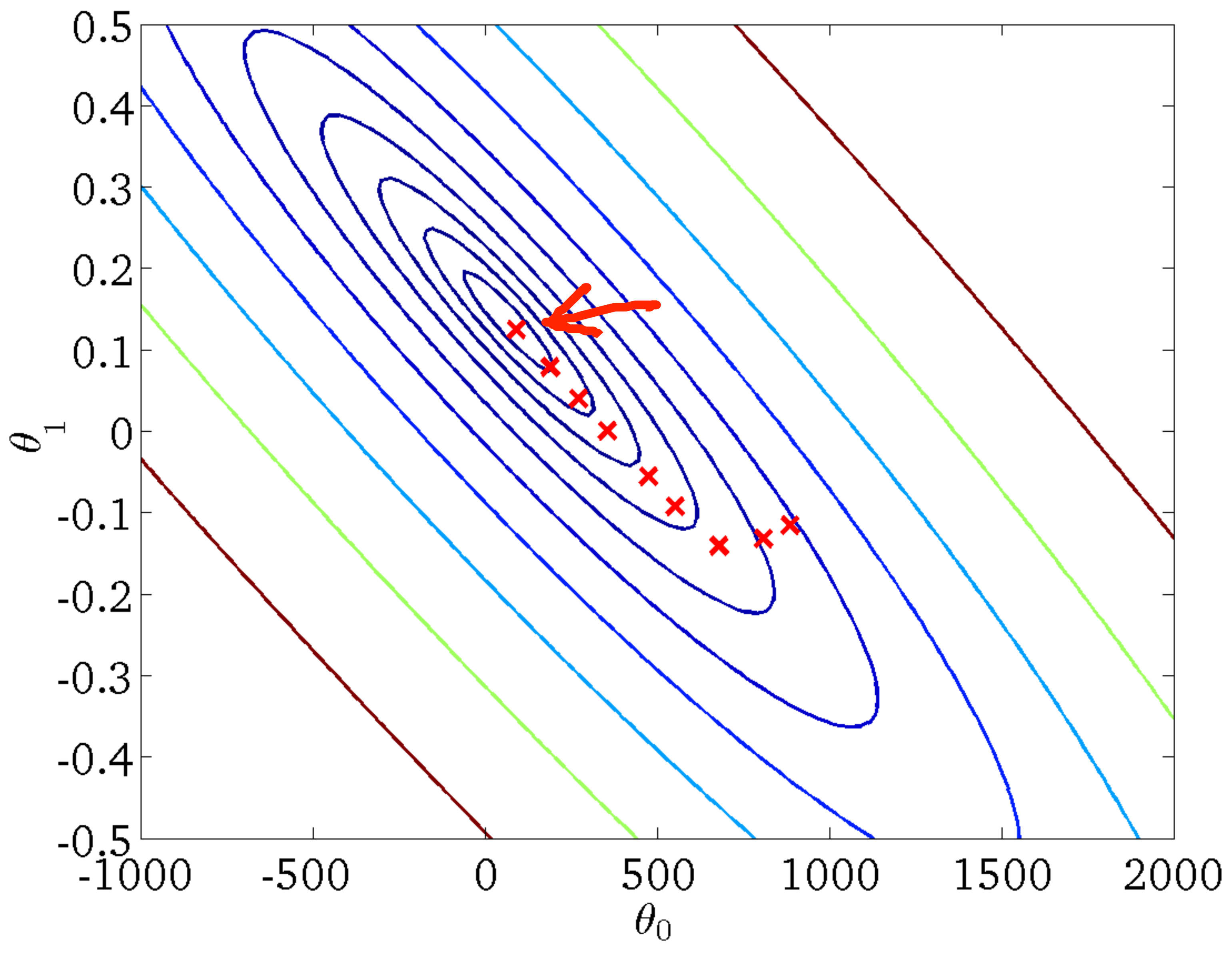
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



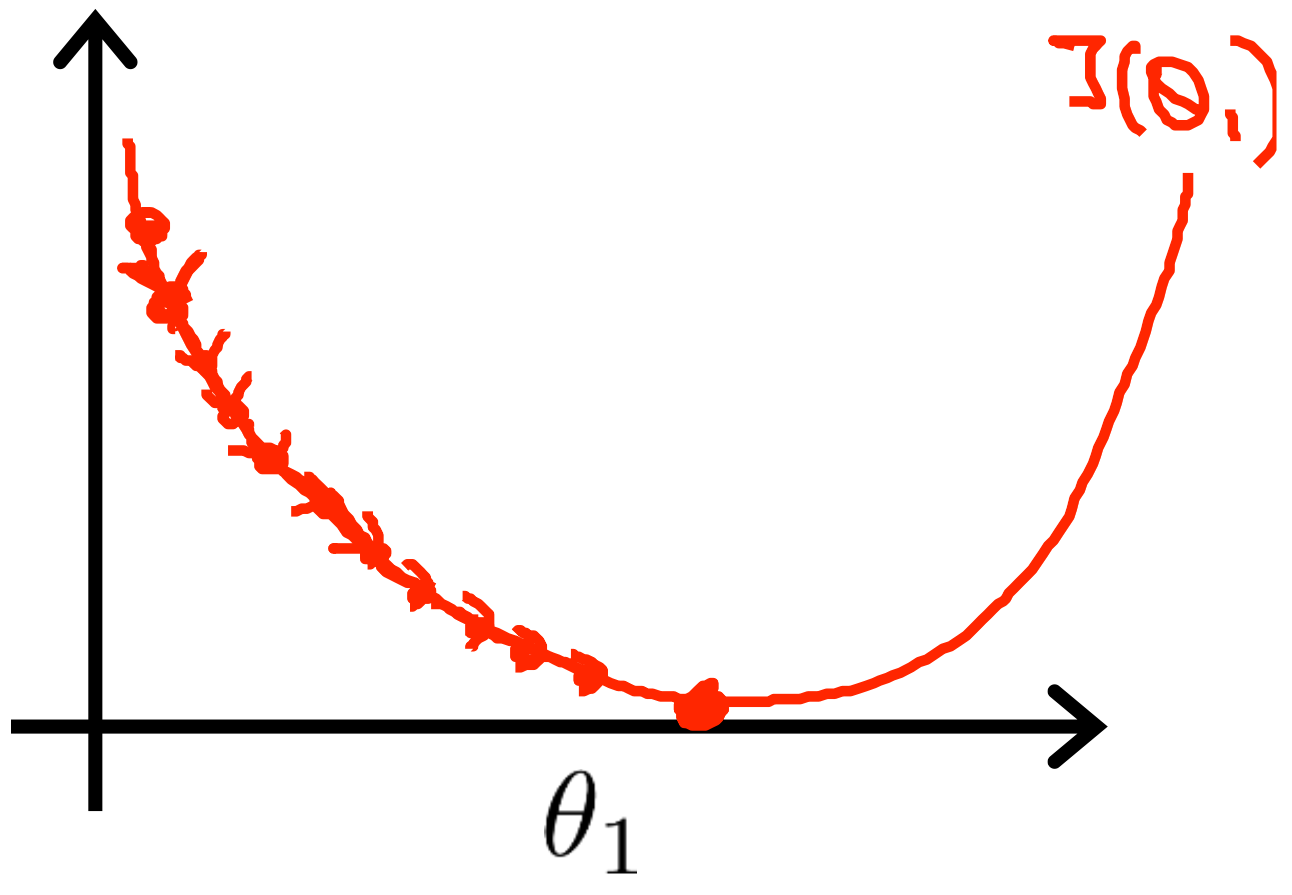
Andrew Ng

Supervised Regression ML

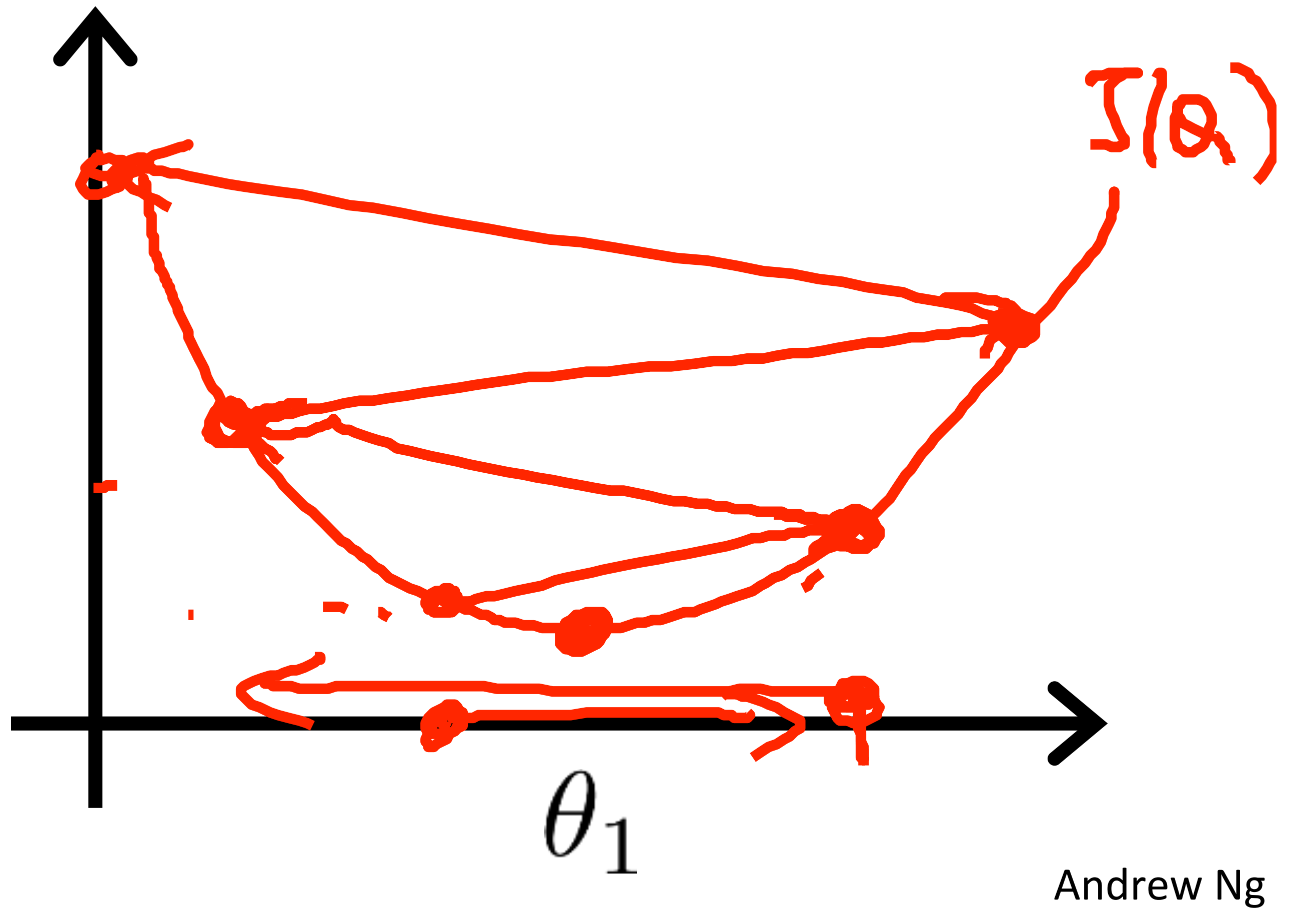
- Gradient Descent (variable increment)

$$\boxed{\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)}$$

If α is too small, gradient descent can be slow.

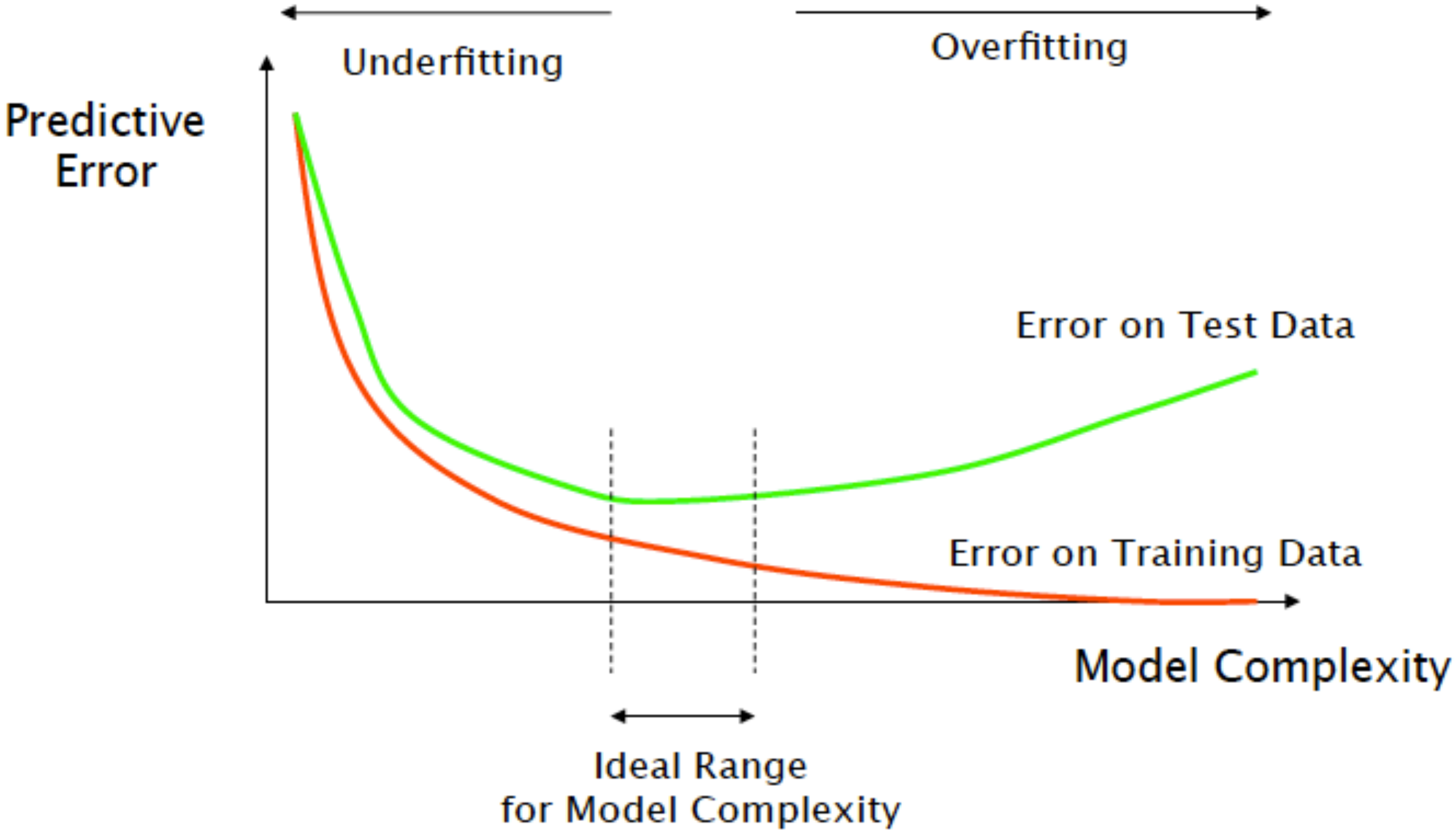


If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



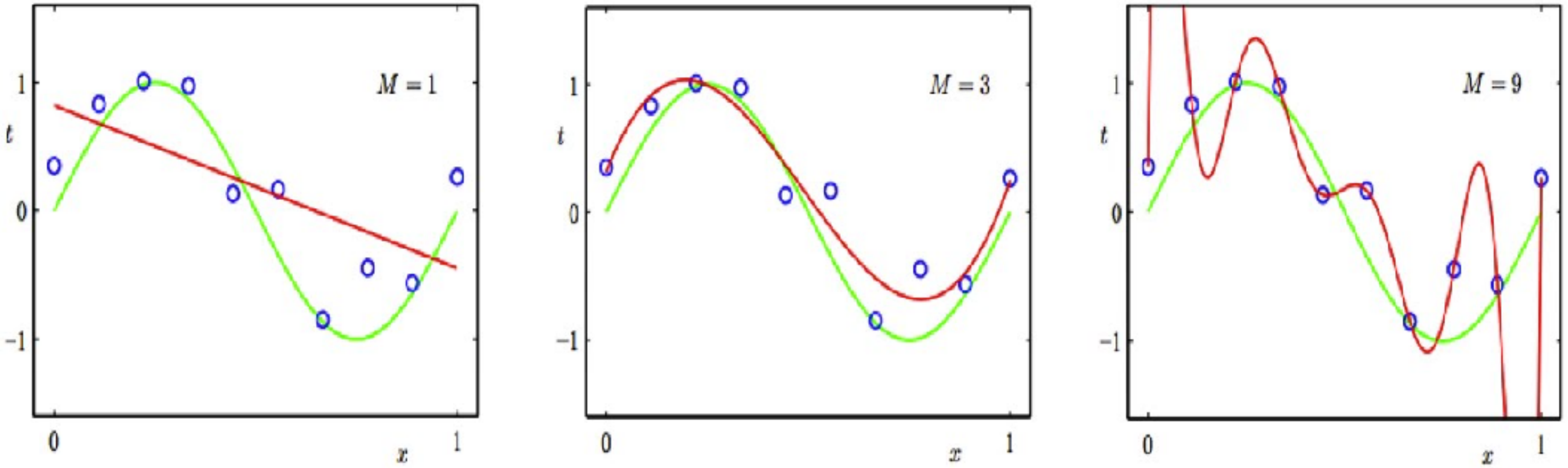
Andrew Ng

Underfitting and Overfitting Problems



Underfitting and Overfitting Examples

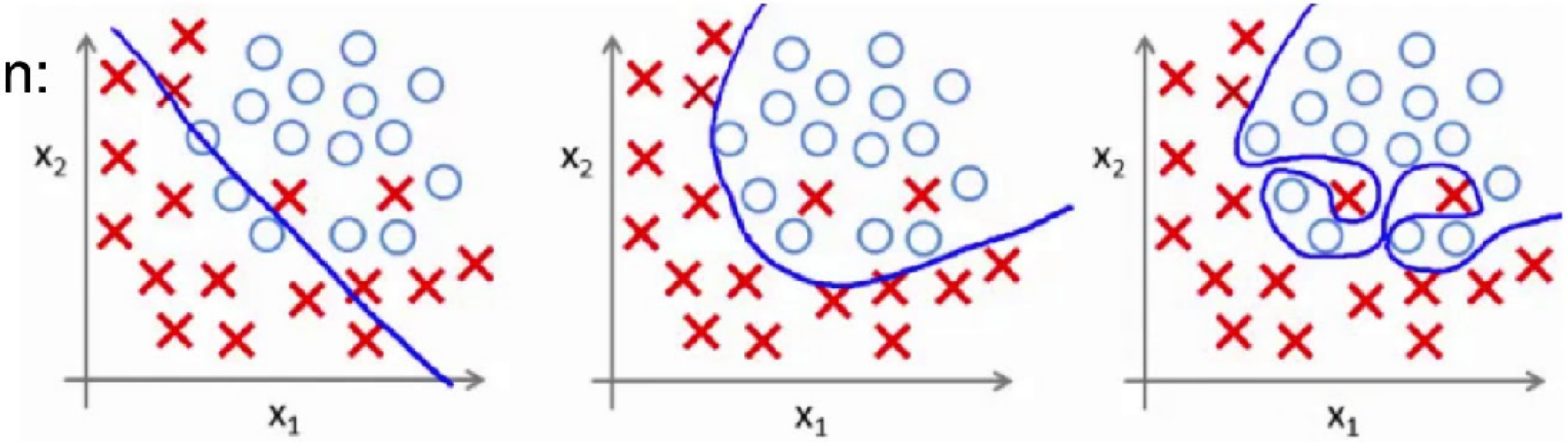
Regression:



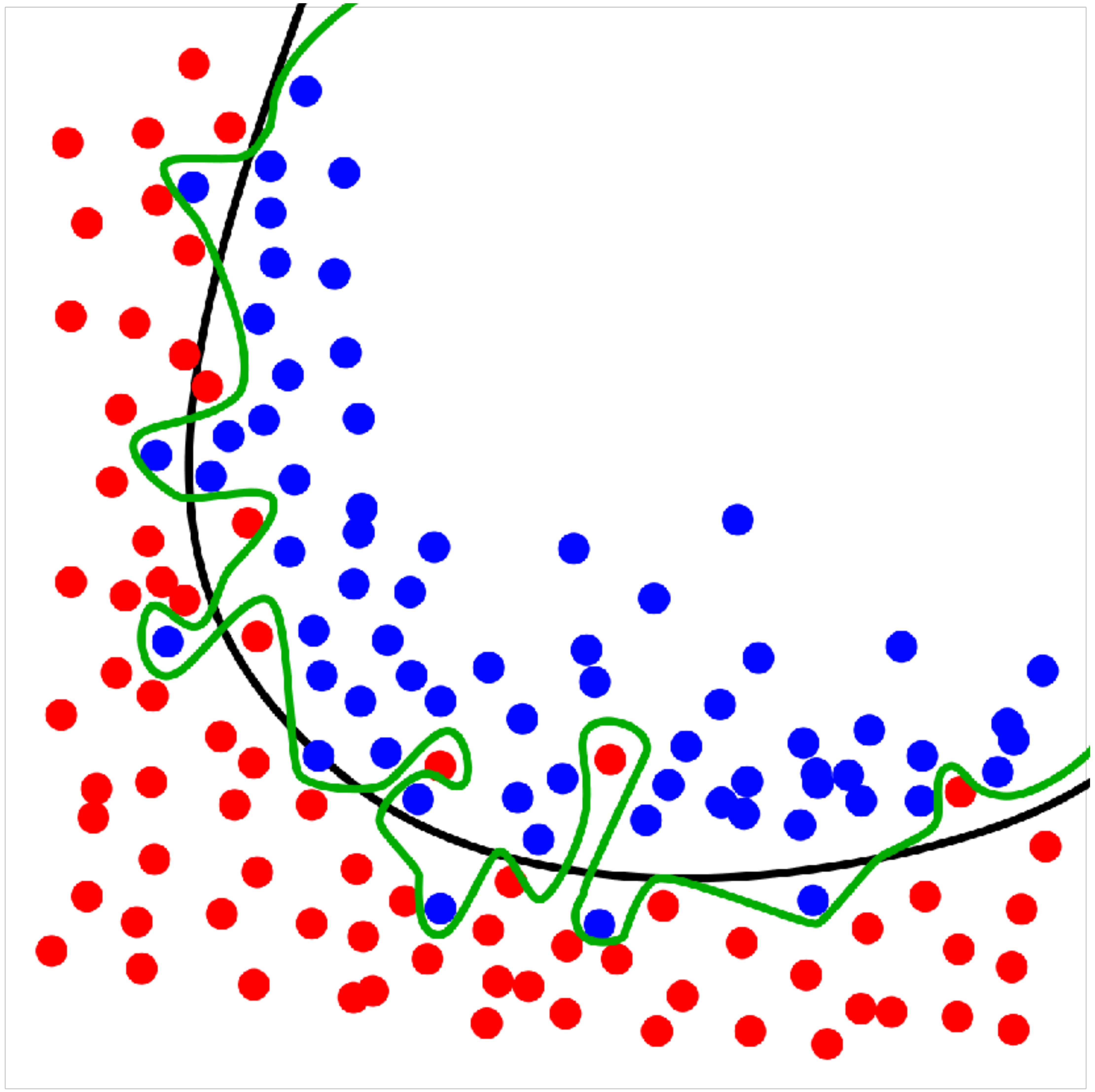
predictor too inflexible:
cannot capture pattern

predictor too flexible:
fits noise in the data

Classification:

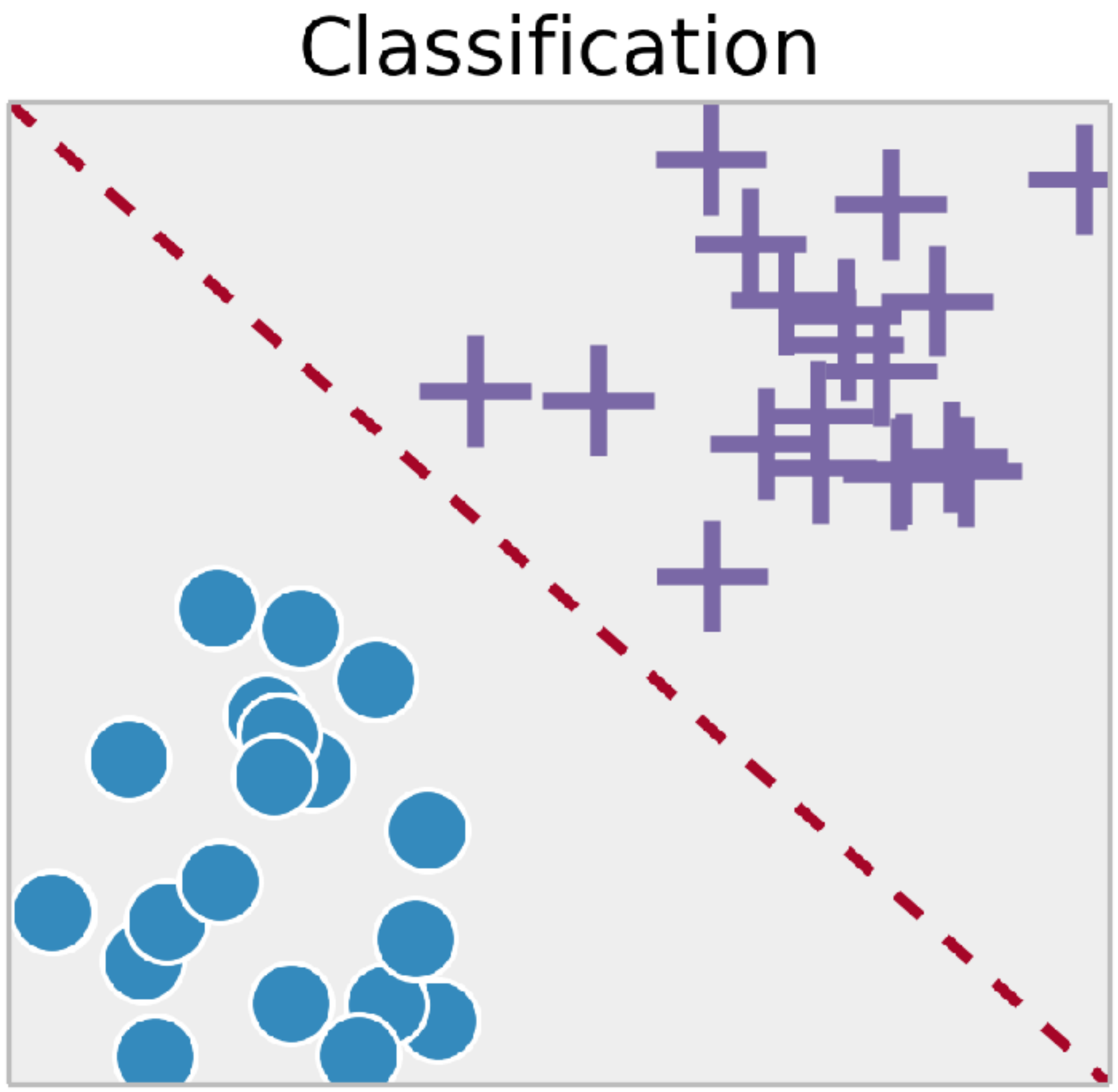


Regularization

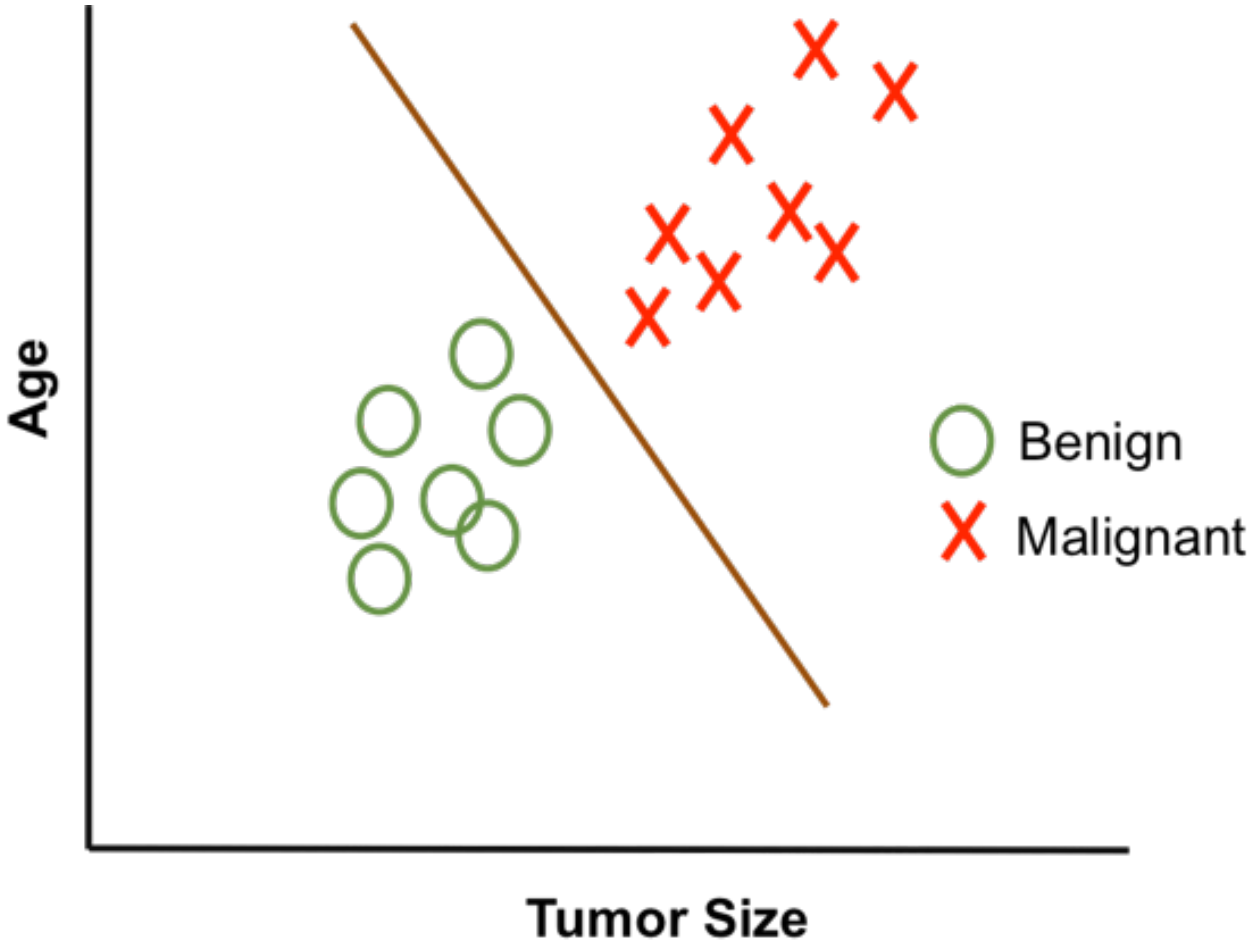


Supervised Machine Learning

Classification



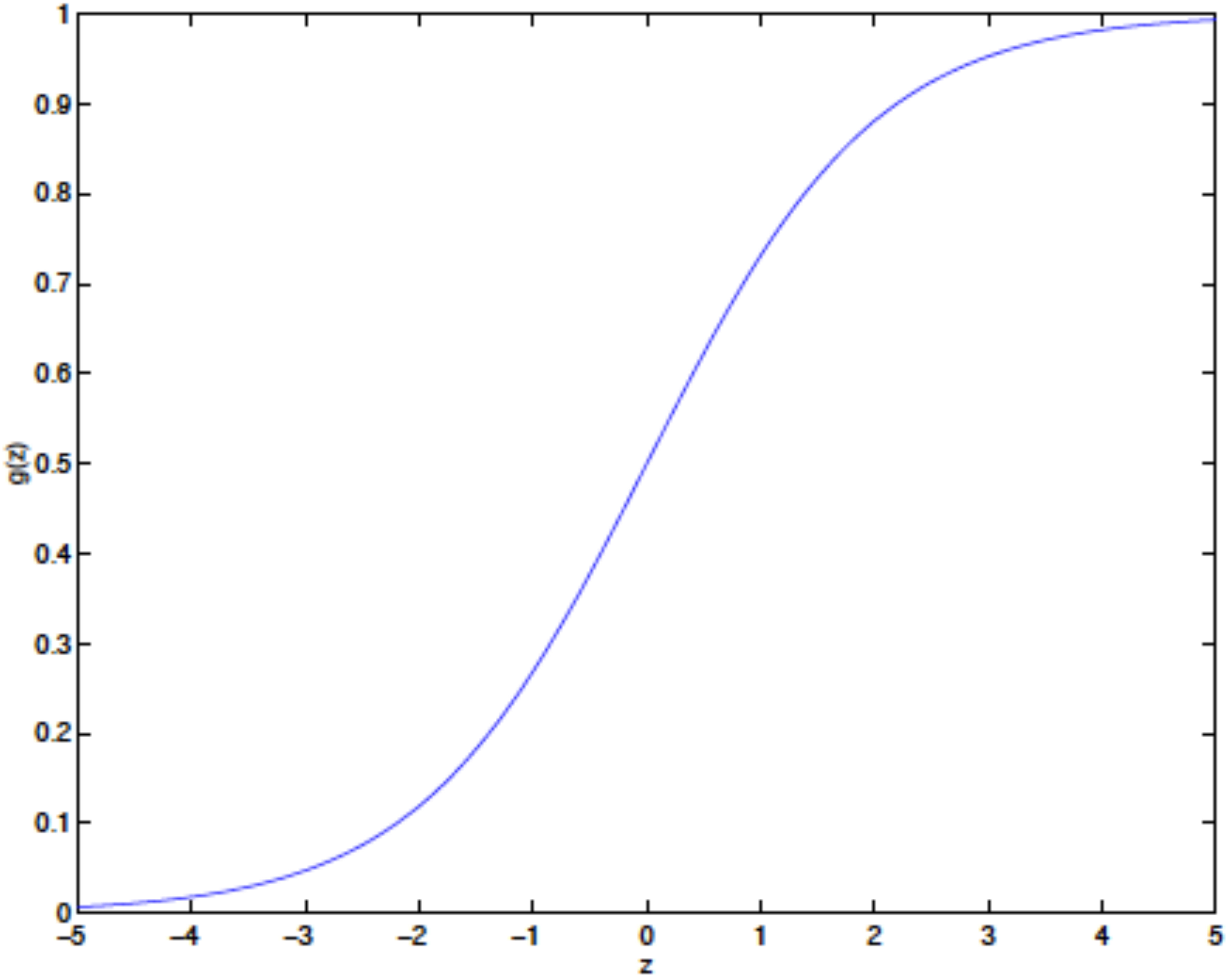
Supervised Classification ML



Supervised Classification ML

- Hypothesis

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

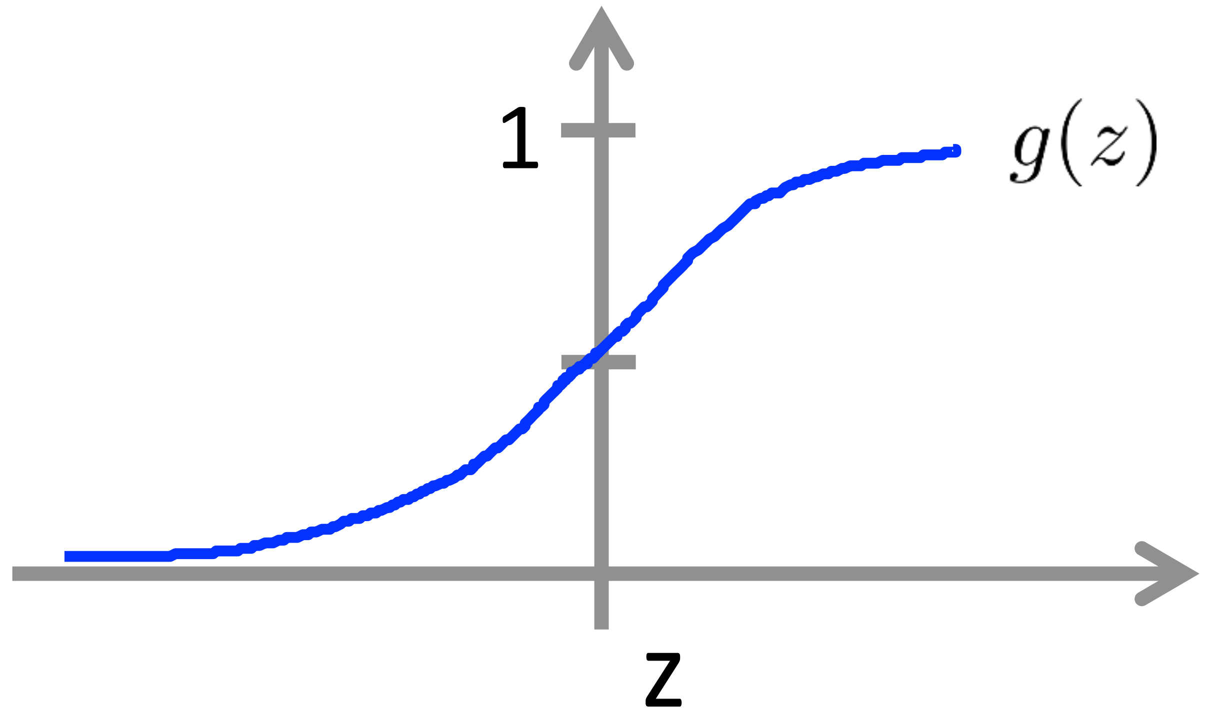


Supervised Classification ML

Logistic regression

$$h_\theta(x) = g(\theta^T x)$$

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



Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

$\theta^T x \geq 0$

$g(z) \geq 0.5$

when $z \geq 0$

$h_\theta(x) = g(\theta^T x$

predict "$y = 0$" if $h_\theta(x) < 0.5$

$\theta^T x < 0$

$g(z) < 0.5$

when $z < 0$

Andrew Ng

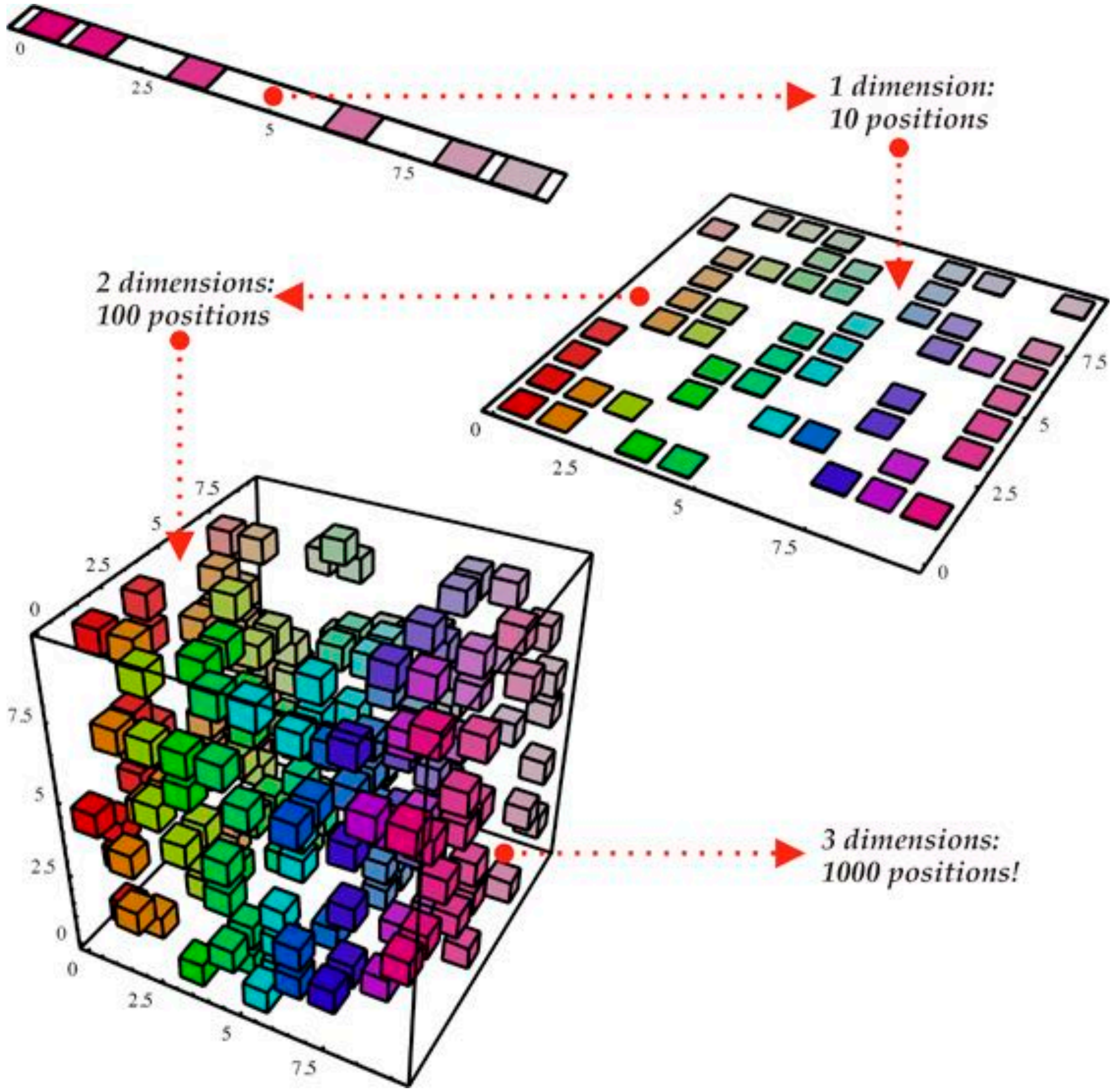
Unsupervised Machine Learning

Unsupervised Machine Learning

- In an unsupervised learning problem, there is no prior information about the problem at had
- The systems needs to learn how to represent a particular input pattern
- Continuous outputs: **dimensionality reduction problem**
- Discrete outputs: **clustering problem**

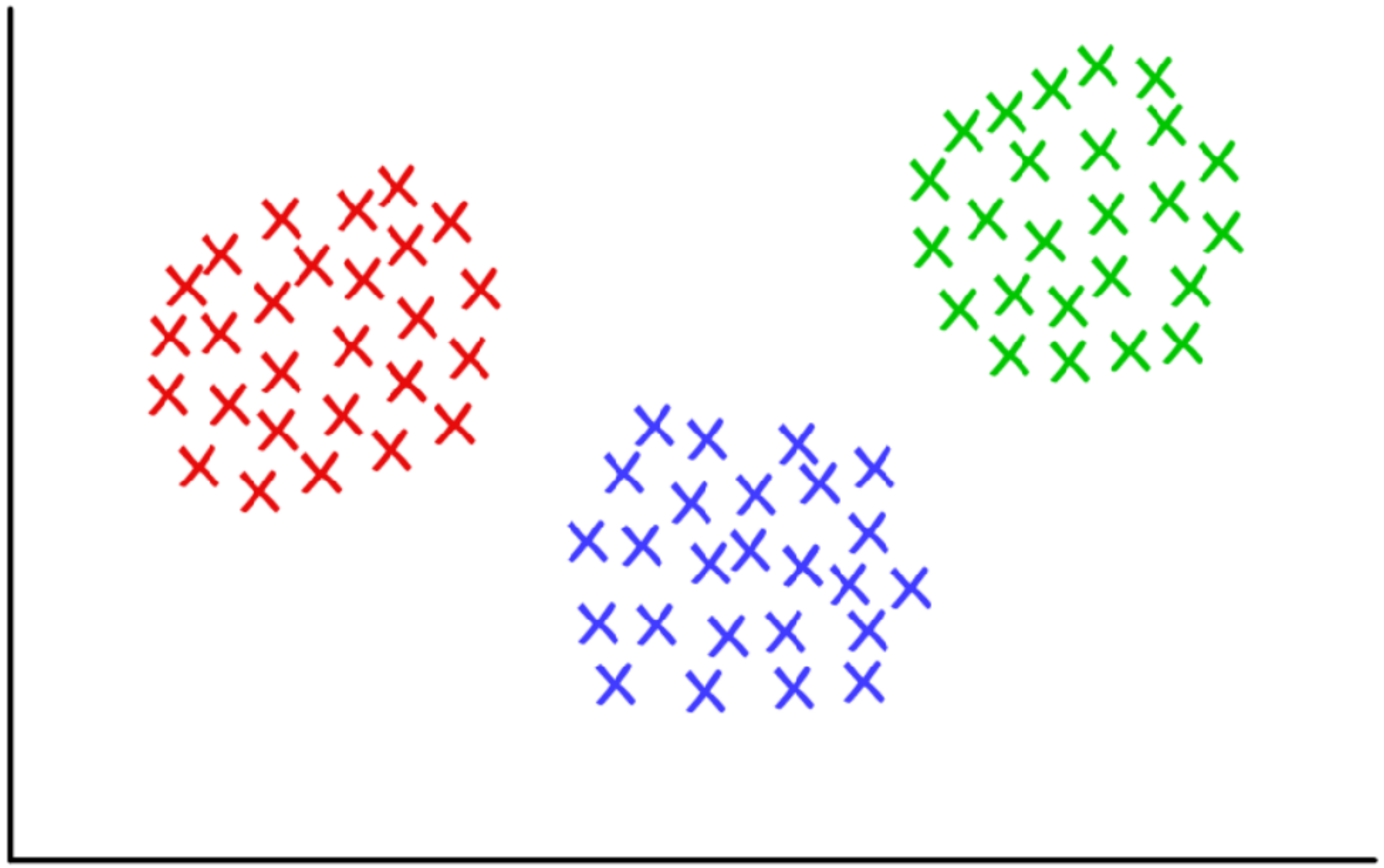
Supervised Machine Learning

- **Dimensionality Reduction**: are used to reduce a N-dimensional dataset to a smaller data i-representation (i.e. N > i > 1)



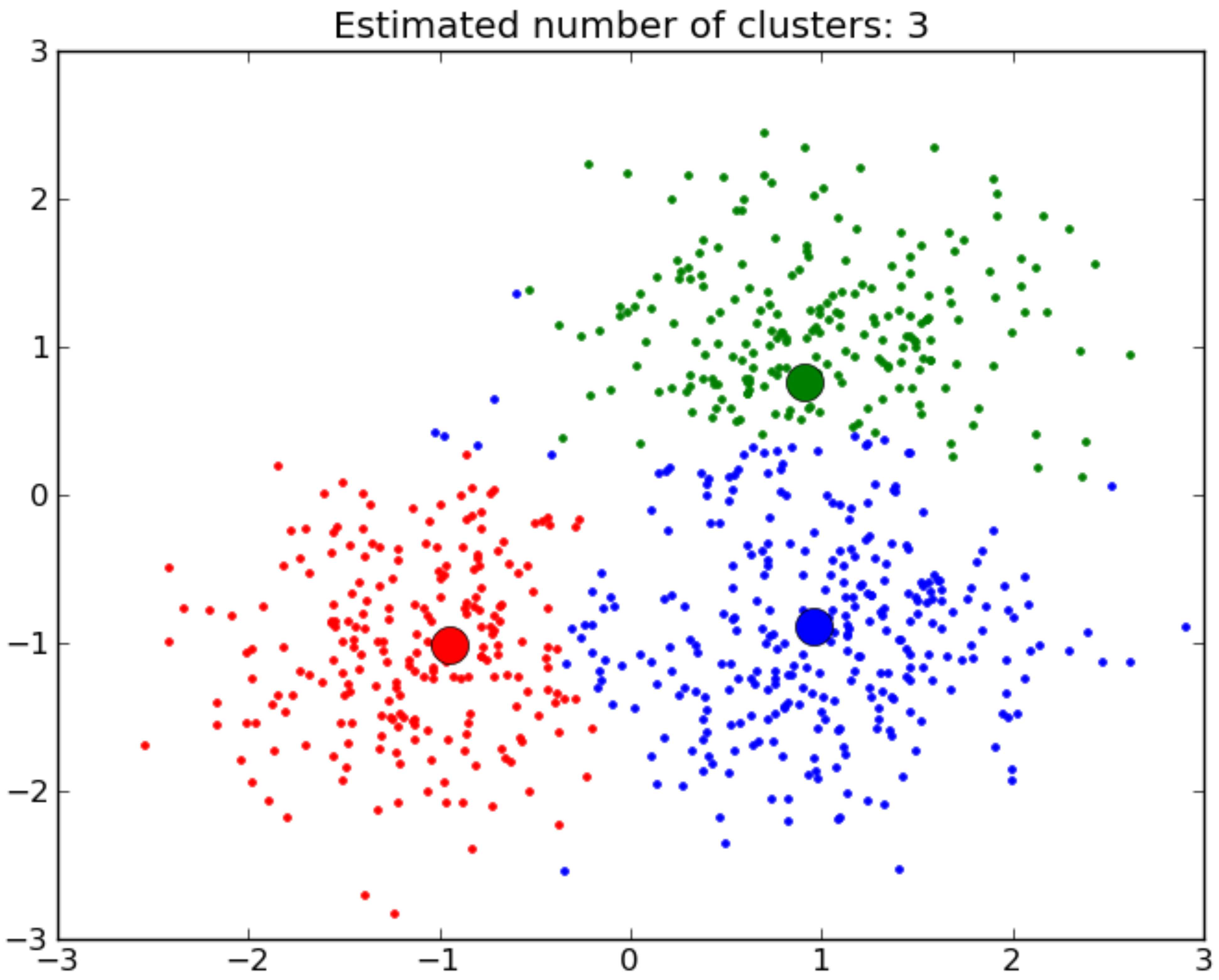
Supervised Machine Learning

- **Clustering**: are used to classify the input data in a set of N-classes

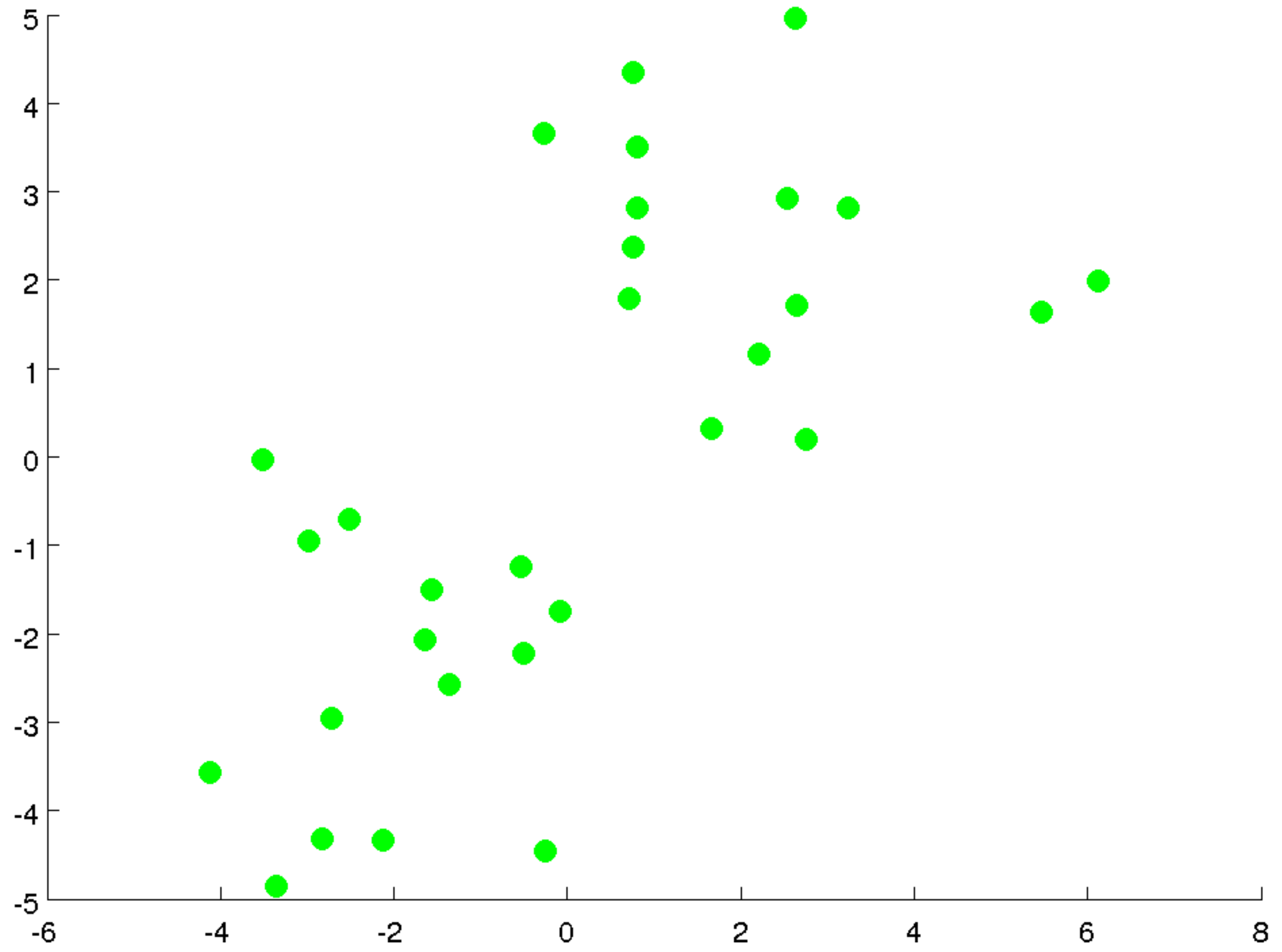


Unsupervised Machine Learning

Clustering

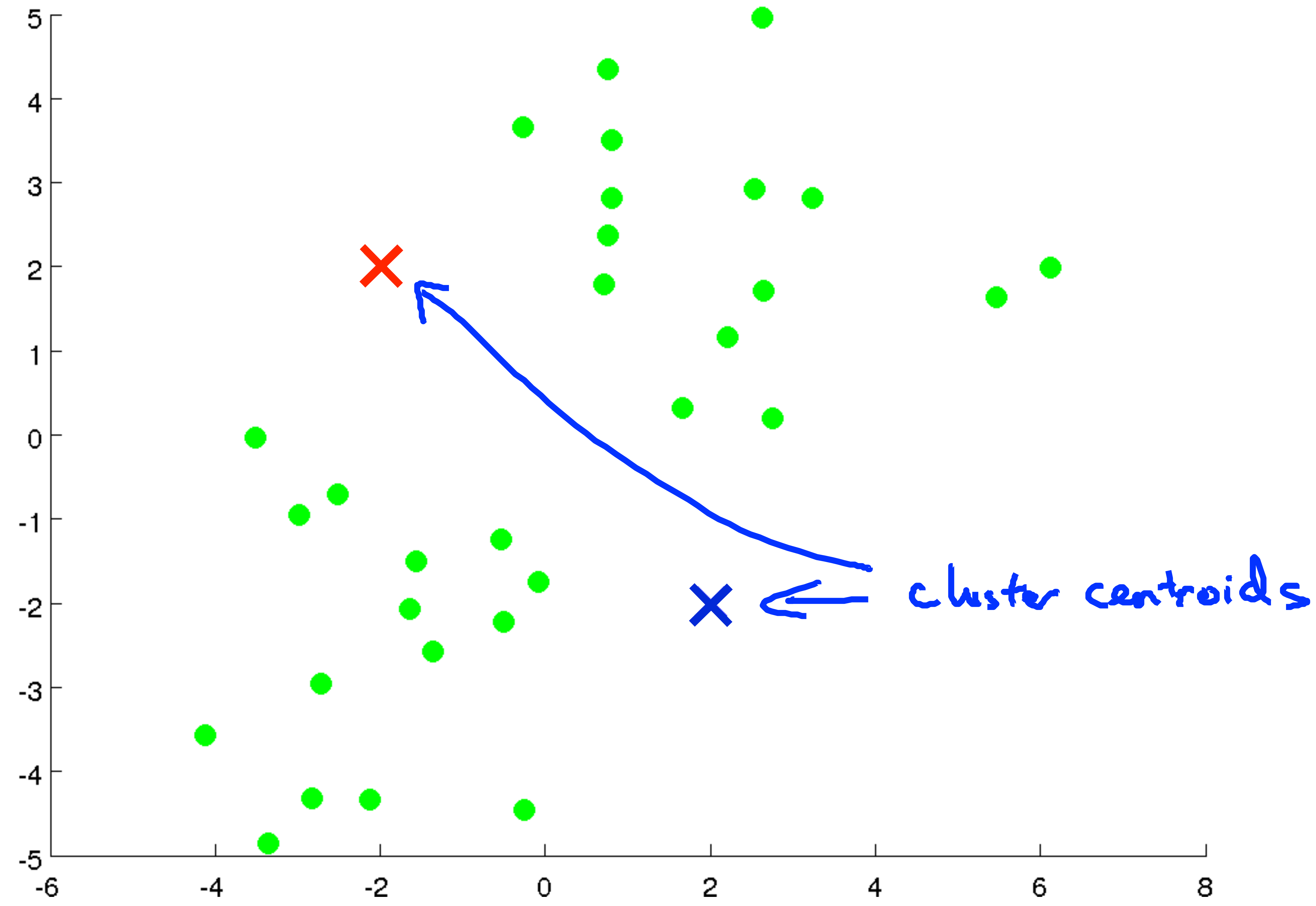


Clustering Dataset Example



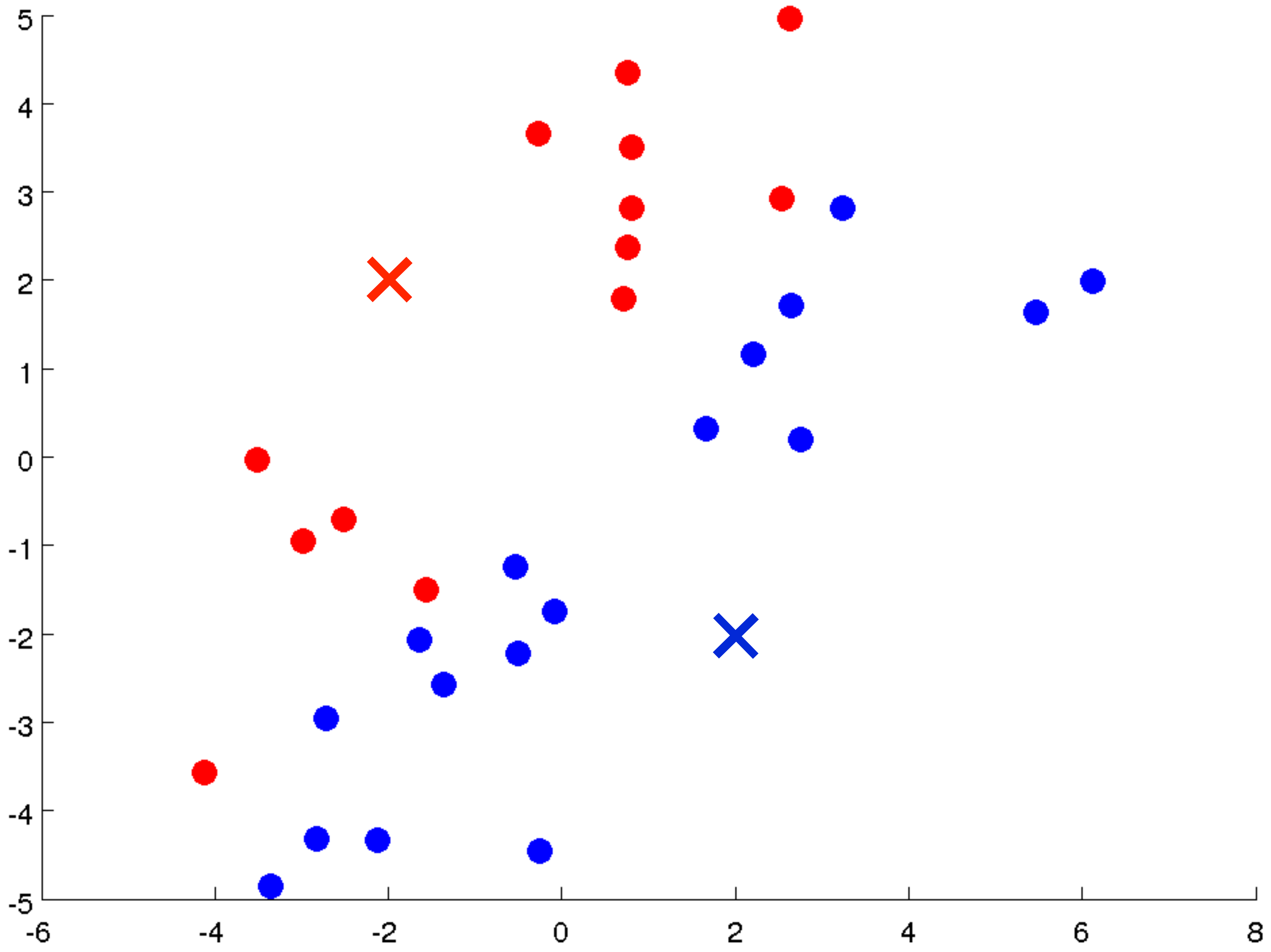
Andrew Ng

Clustering (K-Means)



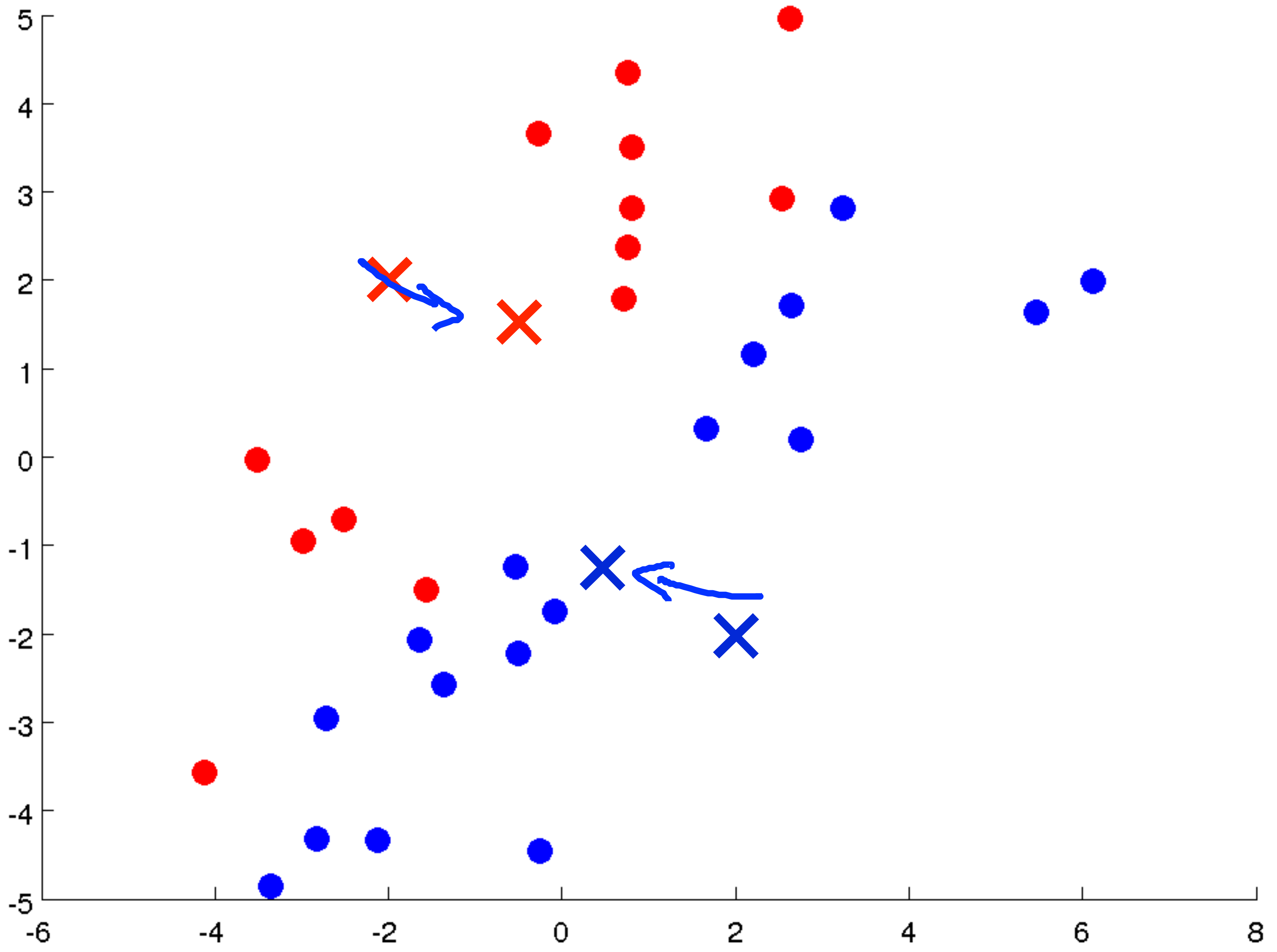
Andrew Ng

Clustering (K-Means)



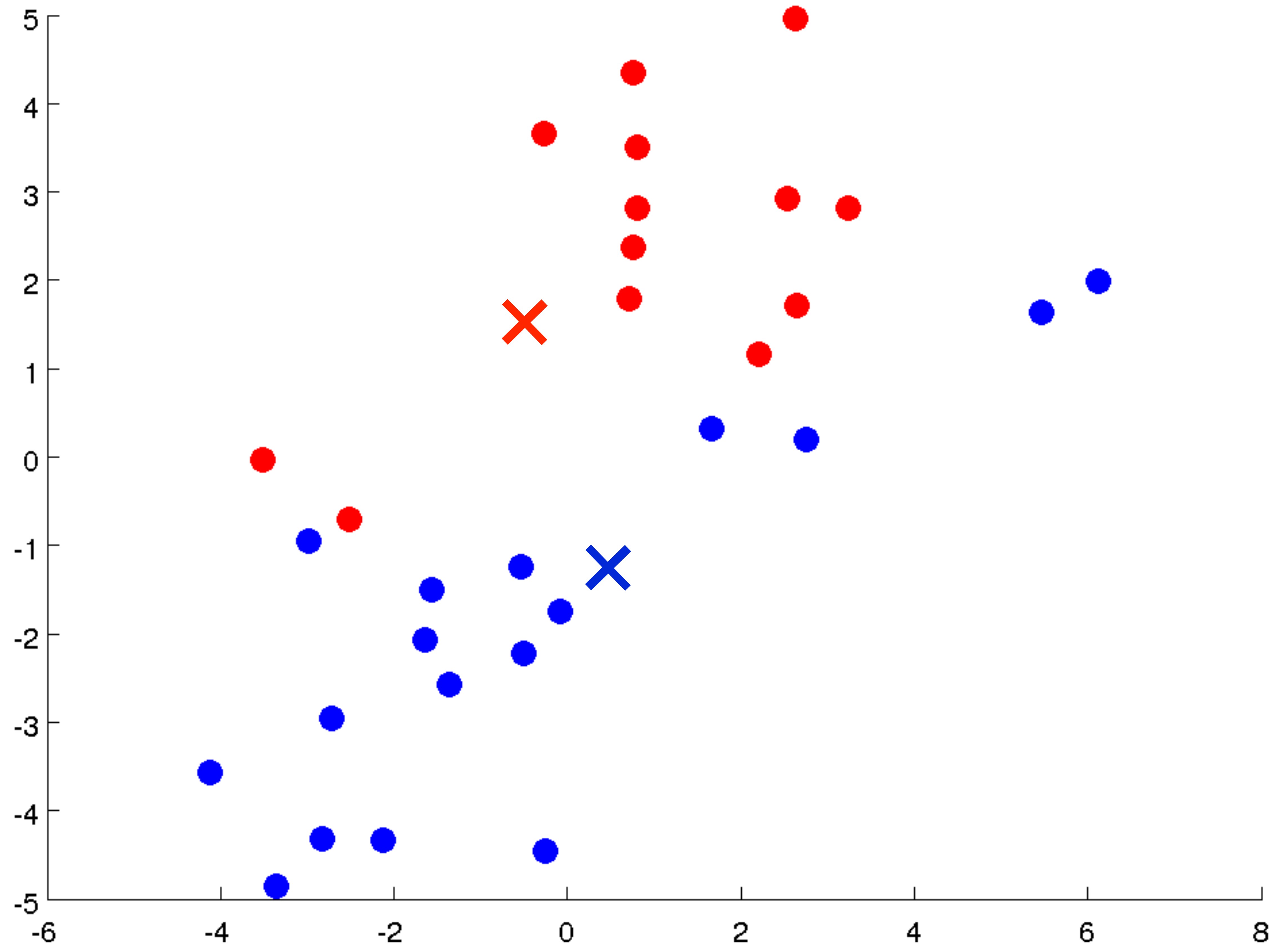
Andrew Ng

Clustering (K-Means)



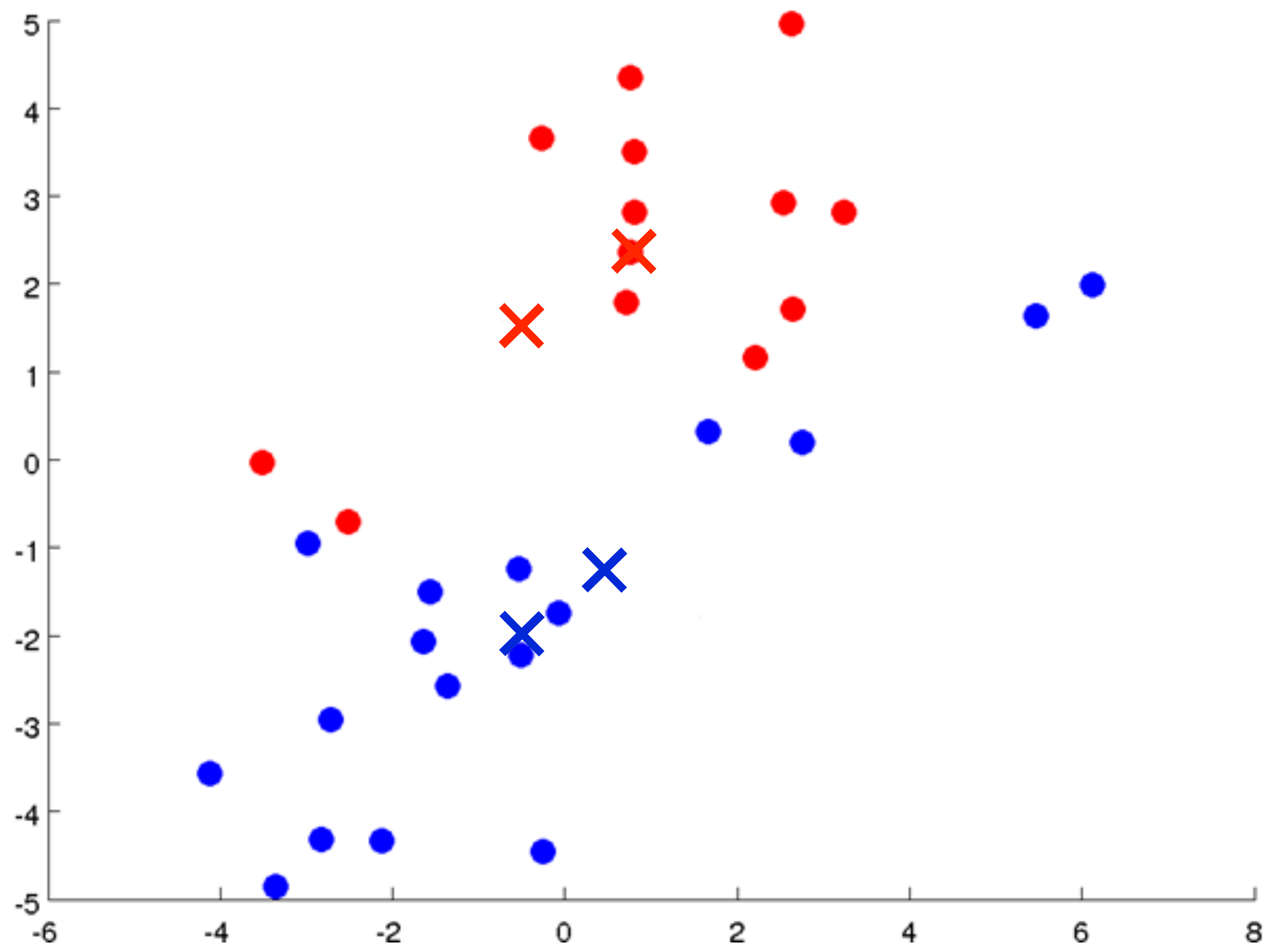
Andrew Ng

Clustering (K-Means)



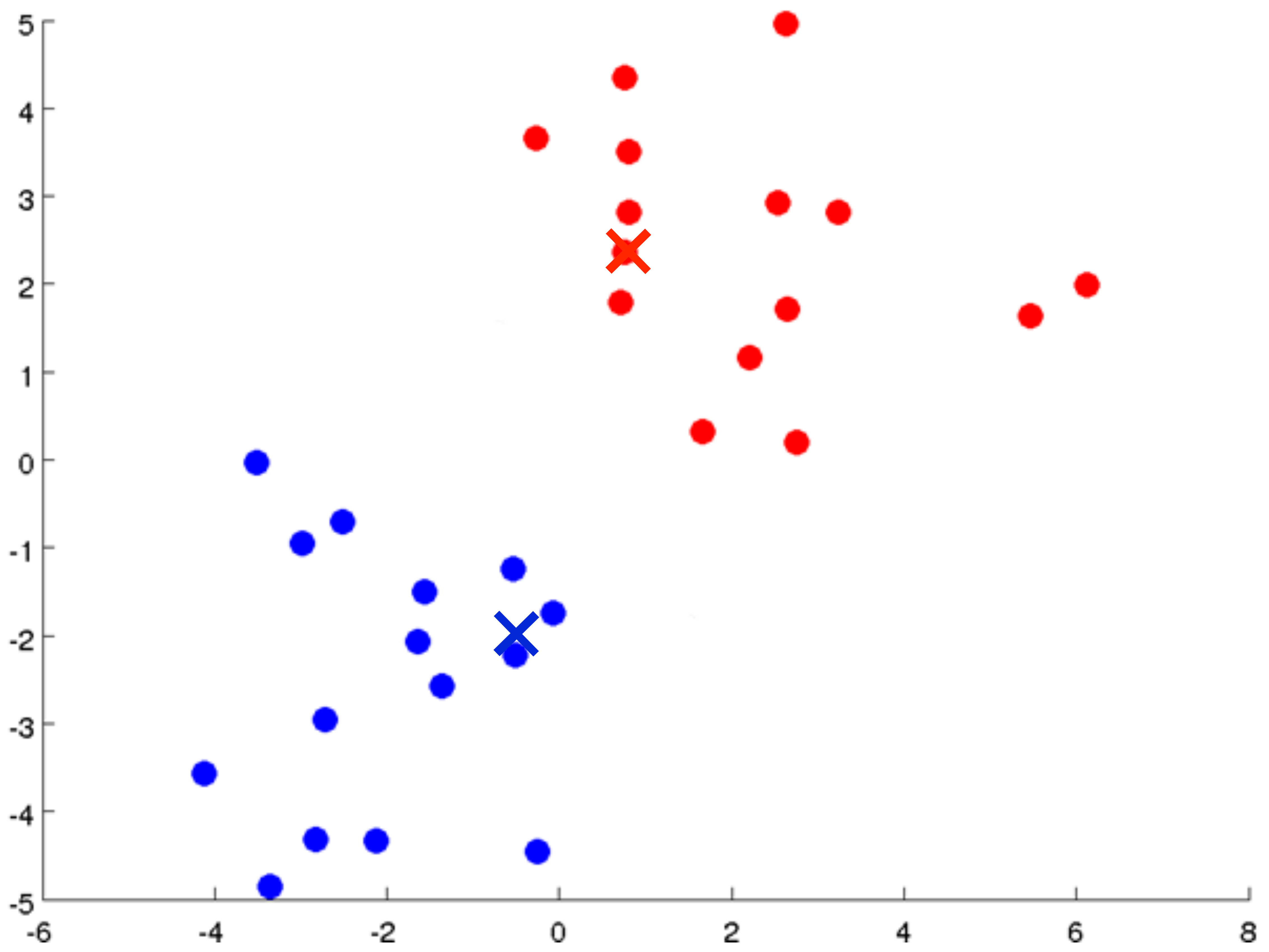
Andrew Ng

Clustering (K-Means)



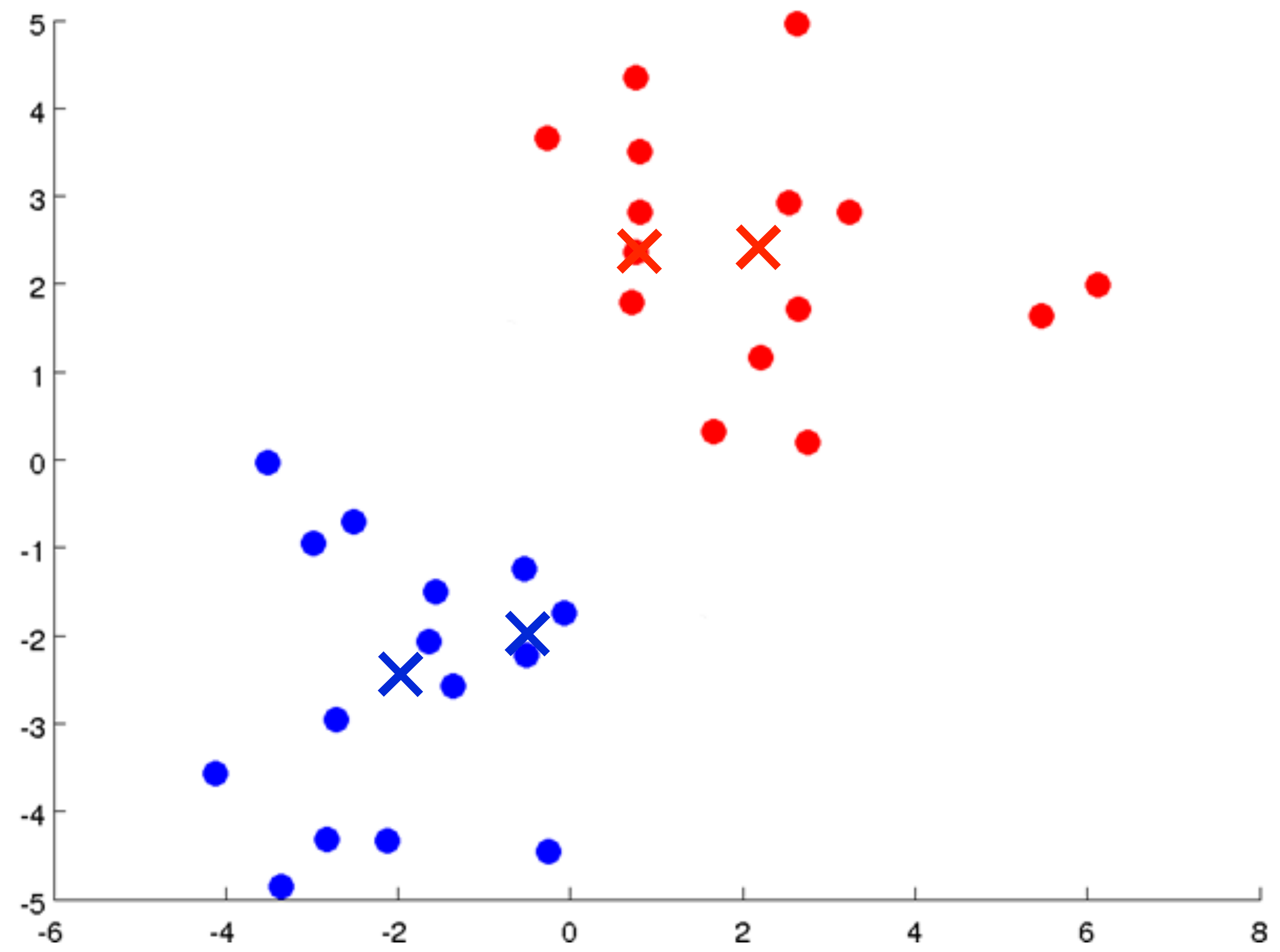
Andrew Ng

Clustering (K-Means)



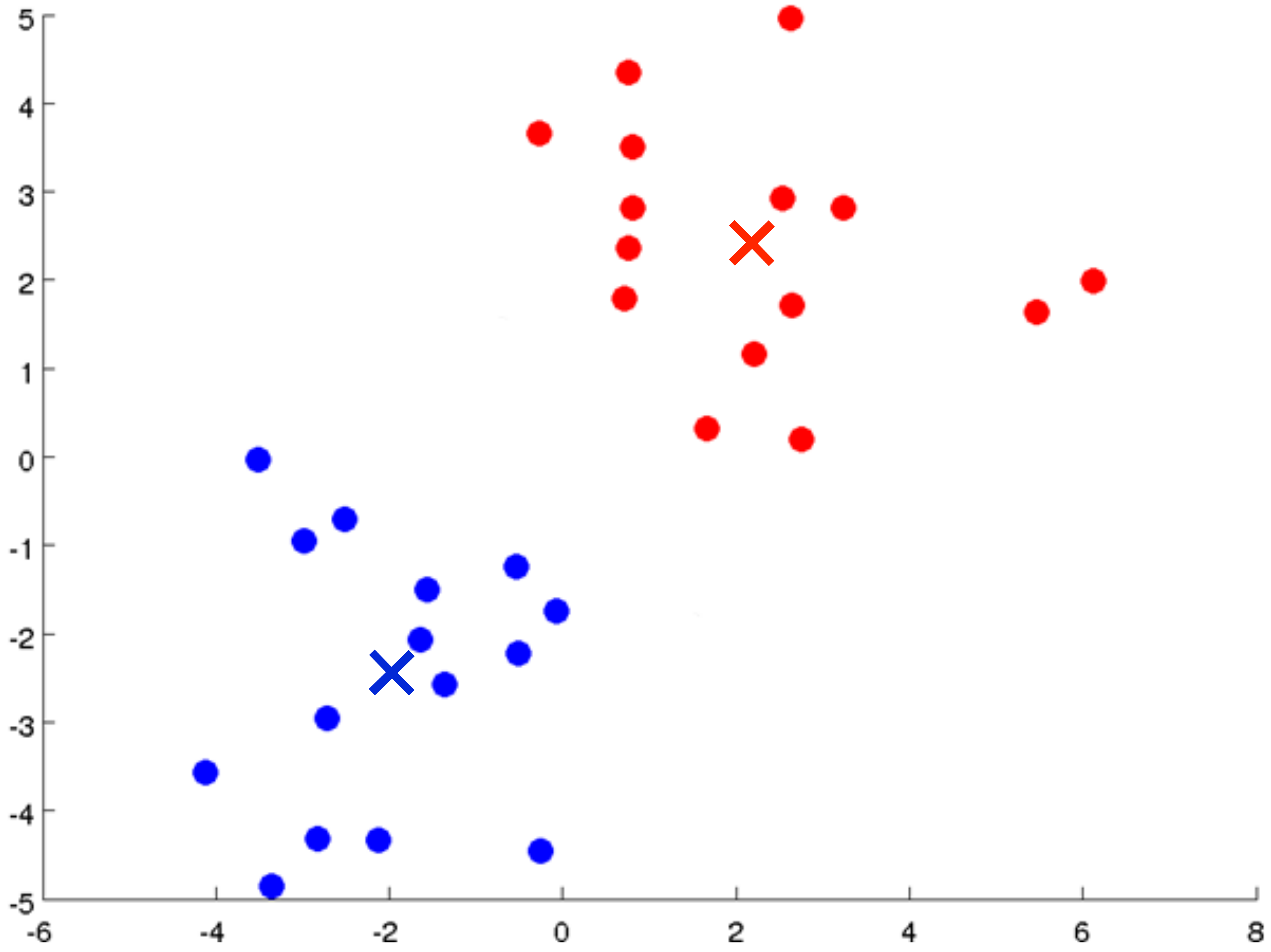
Andrew Ng

Clustering (K-Means)



Andrew Ng

Clustering (K-Means)



Andrew Ng

K-Means Pseudocode

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.

2. Repeat until convergence: {

   For every i, set

   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

   For each j, set

   $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$

   }

References

- Machine Learning Course
  - Stanford University by Coursera (Andrew Ng)
- Introduction to Image Analysis and Machine Learning
  - IT University of Copenhagen (Dan Witzner Hansen)