# Introduction

Introduction to machine learning

# Machine learning?

- Learning from data
  - Large datasets, from the growth of the internet, medical records, cameras & images are ubiquitous, …
- Applications we can’t program by hand
  - Handwriting recognition, NLP, Computer Vision, …
- «Self-learning» algorithms
  - e.g. product or movie recommendations, spam filtering (with occasional/optional supervision input)

# Machine learning?

- Supervised learning
  - Classification, regression
- Unsupervised learning
  - Clustering, dimensionality reduction, density estimation
- Others: Reinforcement learning, sequence learning, semi-supervised learning, …

# Supervised learning - Classification

Cancer data (malignant, benign)

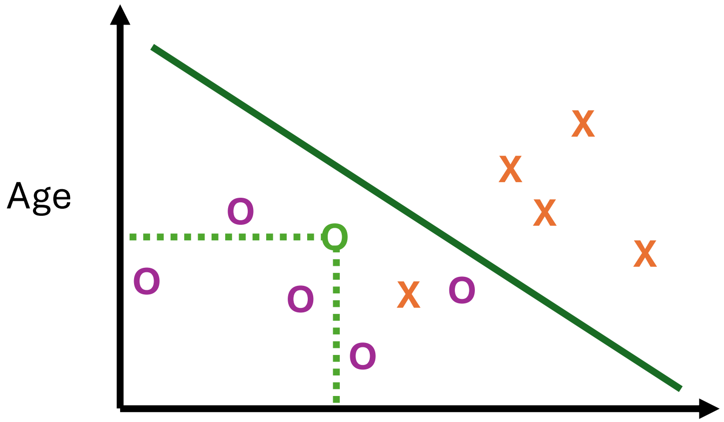
Tumor Size

Discrete output

X = **Malignant = 1**

O = **Benign = -1**

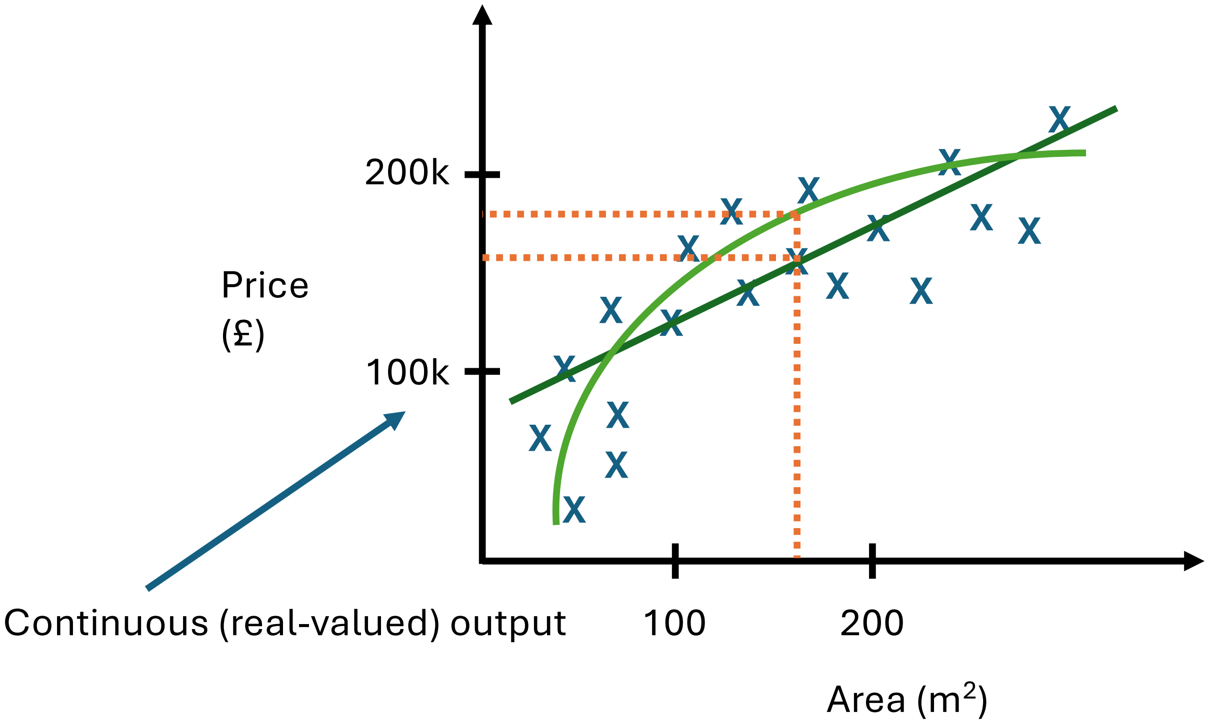
(We could also have more than two output classes – this would be called *multi-class classification*.)

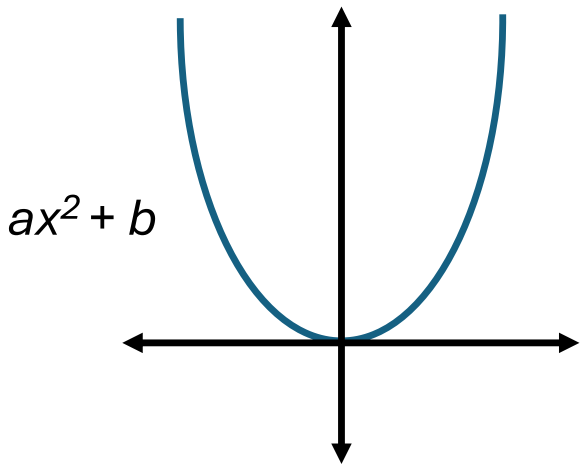



Tumor Size

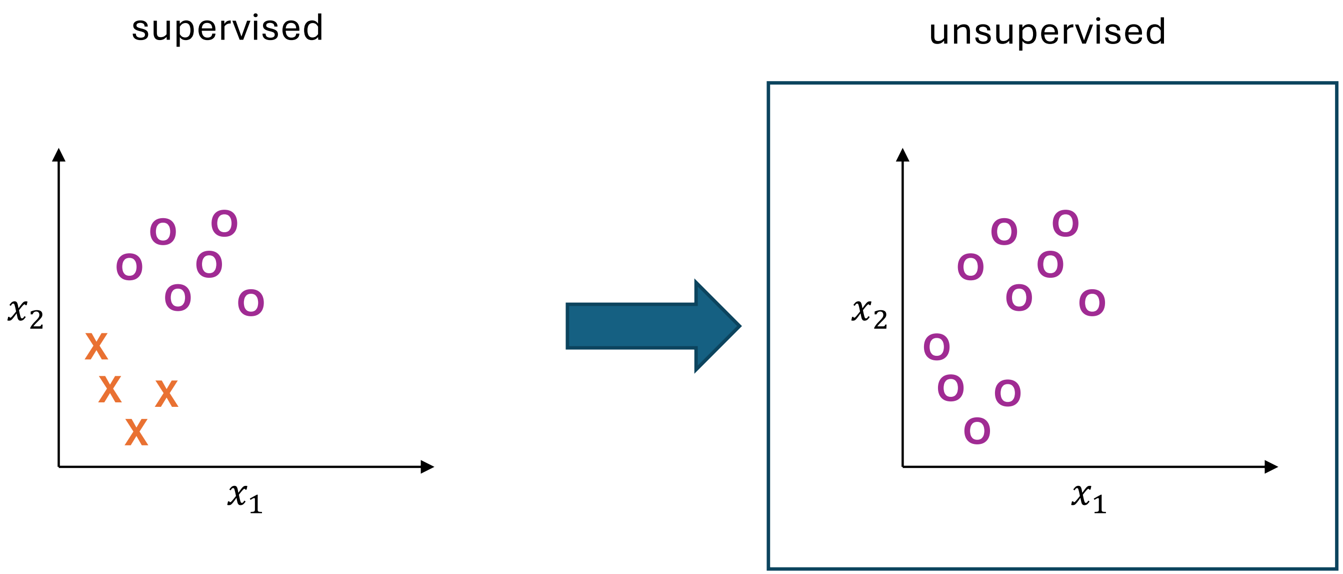
Often, we have more than two input features. Here, that additionally could be tumor clump thickness, uniformity of cell size, uniformity of cell shape, etc.

# Supervised learning - Regression

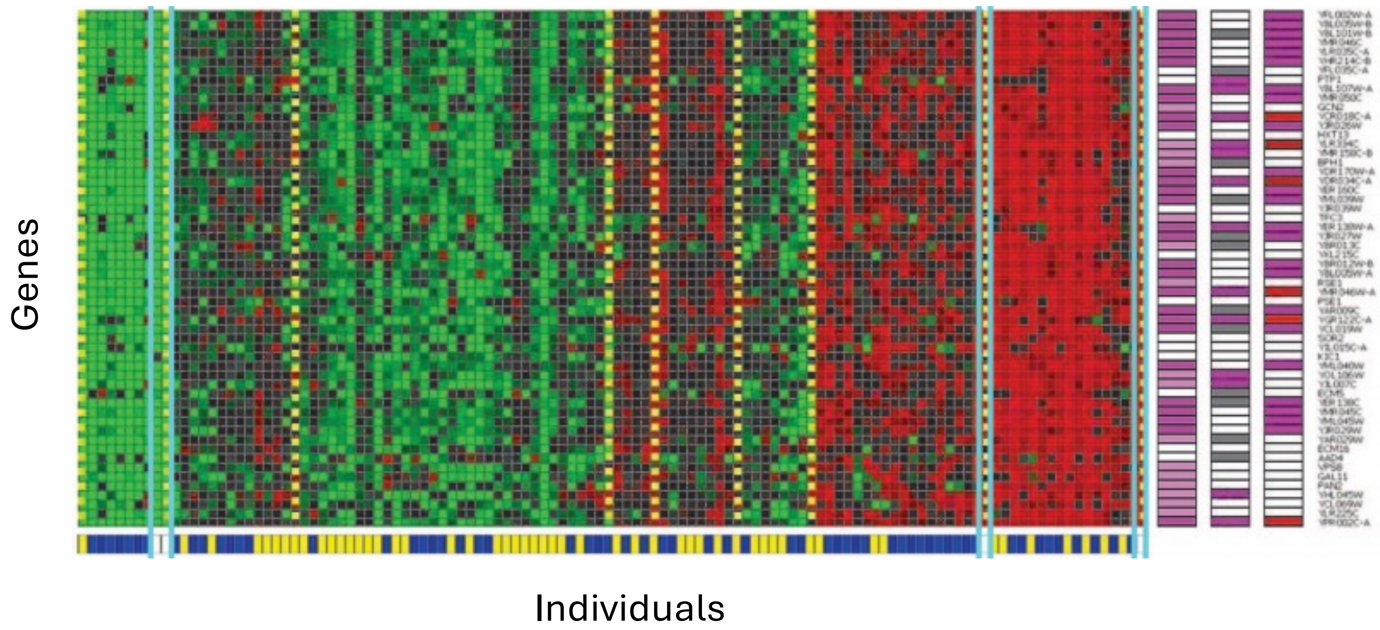


# Unsupervised learning

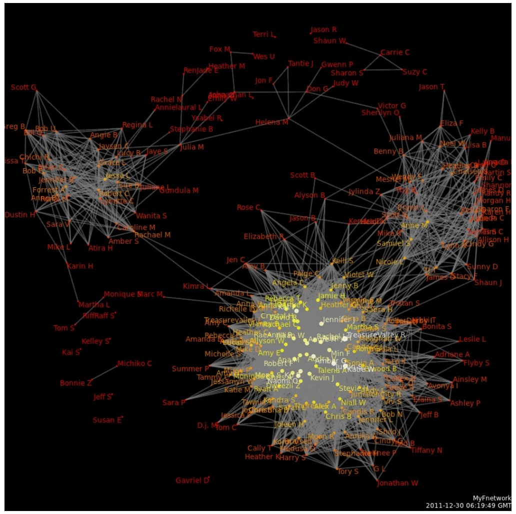


# Unsupervised learning



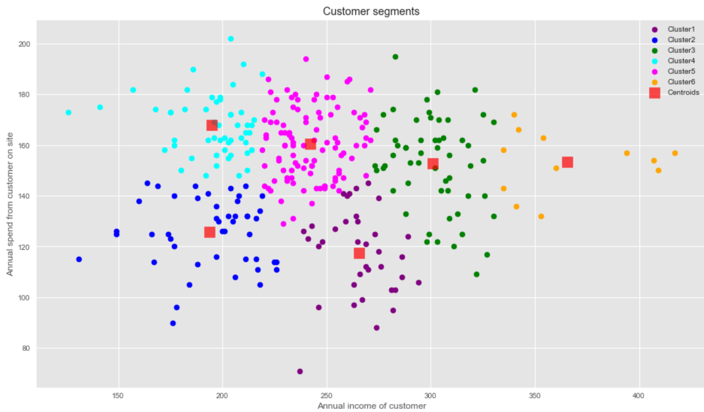

Source: Su-In Lee et al., PNAS 2006

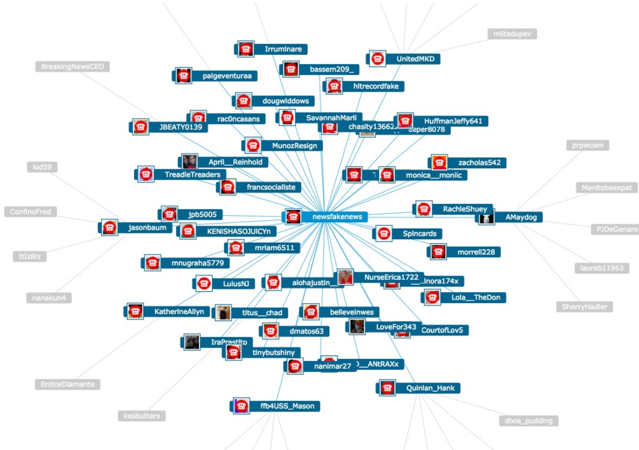
# Unsupervised learning



Social network analysis

Market / customer segmentation



Identifying fake news

Sources:
https://en.wikipedia.org/wiki/Social_network_analysis#/media/File:Kencf0618FacebookNetwork.jpg
https://towardsdatascience.com/clustering-algorithms-for-customer-segmentation
https://medium.com/hackernoon/the-fake-news-arms-race-448675592803

# Machine learning – A magic box?

- Data
- Space of possible solutions
- Characterise objective
- Find algorithm
- Run
- Validate result

# Supervised Learning

# Data

Input Output Number of examples

- Dataset: $\mathcal{D}_n = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$

- $x^{(i)} \in \mathbb{R}^d$ $\quad\quad y^{(i)} \in \{+1, -1\}$ Binary Classification

- $\varphi(x)$: feature representation $\in \mathbb{R}^d$

# Hypotheses

- A hypothesis: $y = h(x; \theta\ \ )$
- $h \in \mathcal{H}$ (hypothesis class)

# Loss function

Guess

- $L(g, a)$

Actual

$$g \in \{+1, -1\}$$

$$a \in \{+1, -1\}$$

- How bad was it that we predicted $g$ when $a$ is the true answer

# Evaluating hypotheses

- Ideally: Small loss on **new** data

$$\mathcal{E}(h) = \boxed{\frac{1}{n'}} \sum_{\boxed{i=n+1}}^{\boxed{n+n'}} L\big(h(x^{(i)}), y^{(i)}\big)$$

test error

- What we can do (for now): Small loss on **training** data

$$\boxed{\mathcal{E}_n(h)} = \boxed{\frac{1}{n}} \sum_{\boxed{i=1}}^{\boxed{n}} L\big(\underbrace{h(x^{(i)})}_{g}, \underbrace{y^{(i)}}_{a}\big)$$

training error

# Learning algorithms


- How to come up with learning algorithms:
  - Be a clever (or not so clever) human
  - Use optimisation methods

# Linear Classifiers

# Linear Classifiers

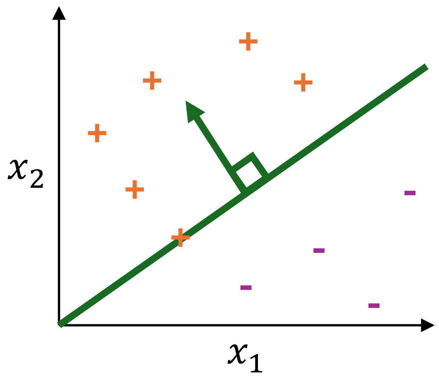
- Linear classifiers: A choice of $\mathcal{H}$

$$h(\overset{\mathbb{R}^d}{x}\,;\overset{\mathbb{R}^d}{\theta},\overset{\mathbb{R}}{\theta_0}) = sign(\underbrace{\overbrace{\theta^T x}^{dot} + \theta_0}_{\mathbb{R}}) = \begin{cases} +1 \text{ if } \theta^T x + \theta_0 > 0 \\ -1 \text{ otherwise} \end{cases}$$

$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ $\quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $\quad \theta : d \times 1$ $\quad x : d \times 1$ $\quad (1 \times d).(d \times 1)$

$\theta_1 . x_1 + \theta_2 . x_2 + \theta_0 = 0$ Implicit representation

$y = \underset{slope}{a}x + b$ Parametric representation



# The random linear classifier algorithm

hyperparameter

random_linear_classifier(D, k):

for j=1 to k

  $\theta^{(j)}$ = random($\mathbb{R}^d$); $\theta_0^{(j)}$ = random($\mathbb{R}$)

j* = $\underset{j \in \{1..k\}}{\operatorname{argmin}} \mathcal{E}_n(\theta^{(j)}, \theta_0^{(j)})$

return($\theta^{(j^*)}, \theta_0^{(j^*)}$)