Imperial College London

Lecture 1: Basic concepts of machine learning

Introduction to machine learning

Kevin Webster

Department of Mathematics
Imperial College London

Outline

- Background
  - Artificial Intelligence, Machine Learning and Deep Learning
  - Example applications
- Machine Learning: the settings
  - ML Tasks
  - Performance Measures & experience
- Machine learning: important concepts
  - Generalisation, overfitting and underfitting
  - Model selection
- Model example: k-nearest neighbours

Outline

Background

- Artificial Intelligence, Machine Learning and Deep Learning
- Example applications

Machine Learning: the settings

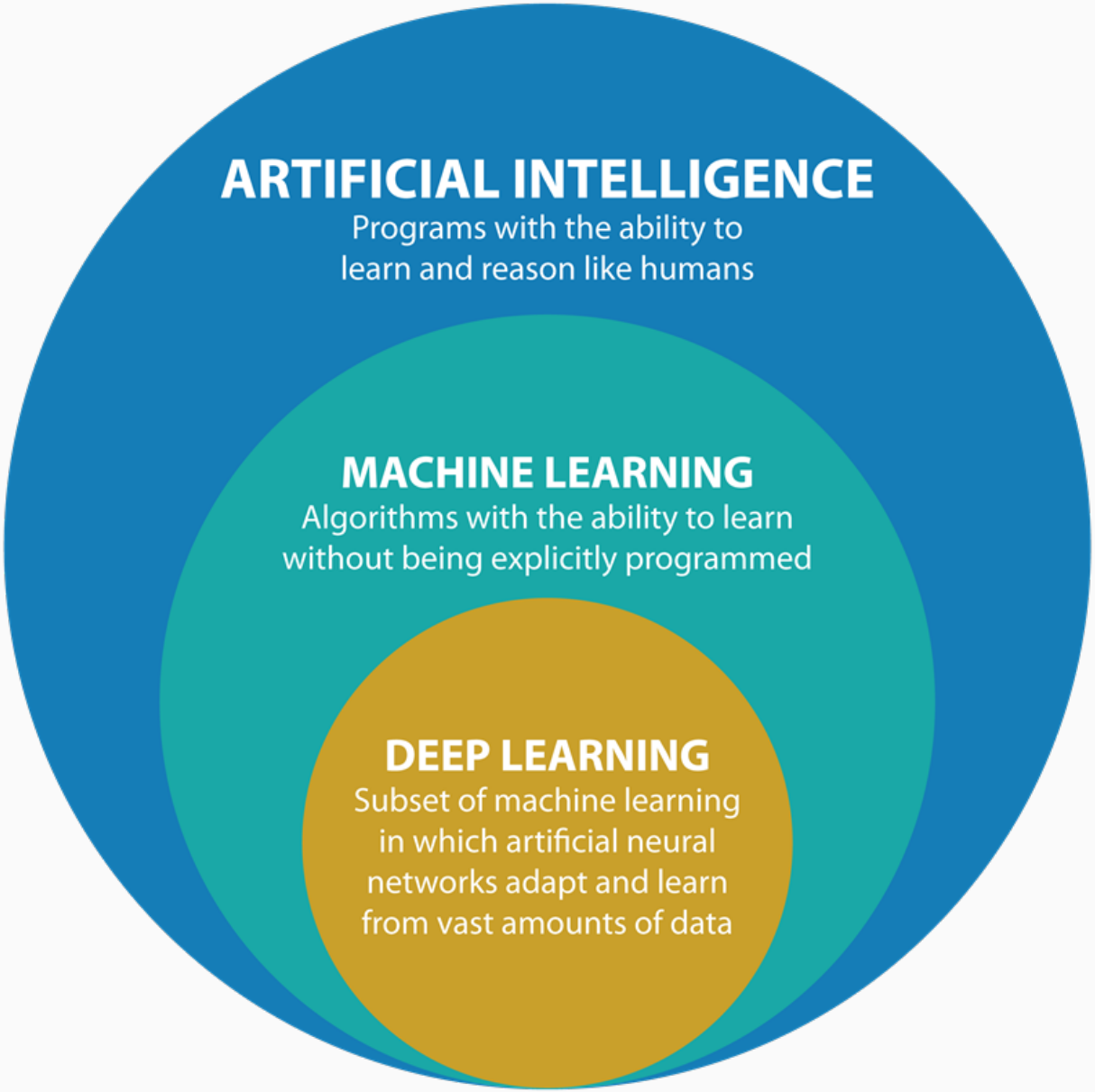
- ML Tasks
- Performance Measures & experience

Machine learning: important concepts

- Generalisation, overfitting and underfitting
- Model selection

Model example: k-nearest neighbours

AI, ML and Deep Learning

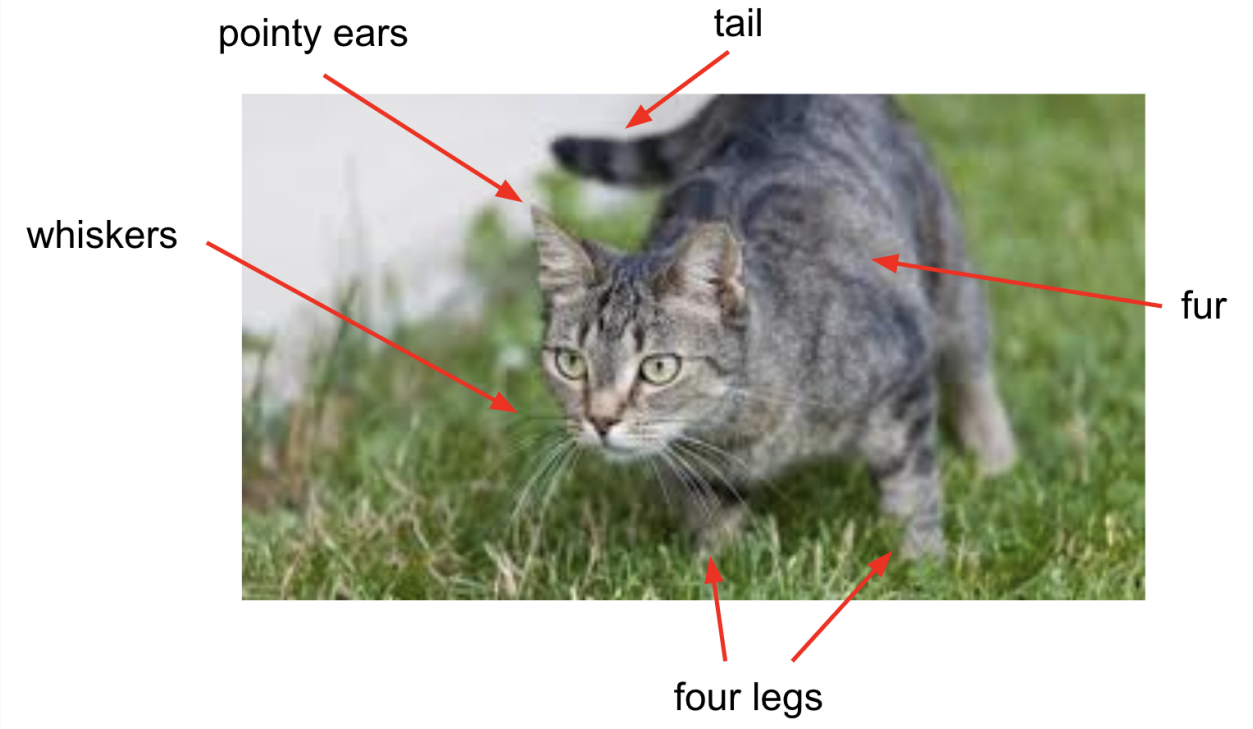


Birth of Artificial Intelligence

'The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.'

- *Proposal from the Dartmouth Conference, 1956*

AI vs ML



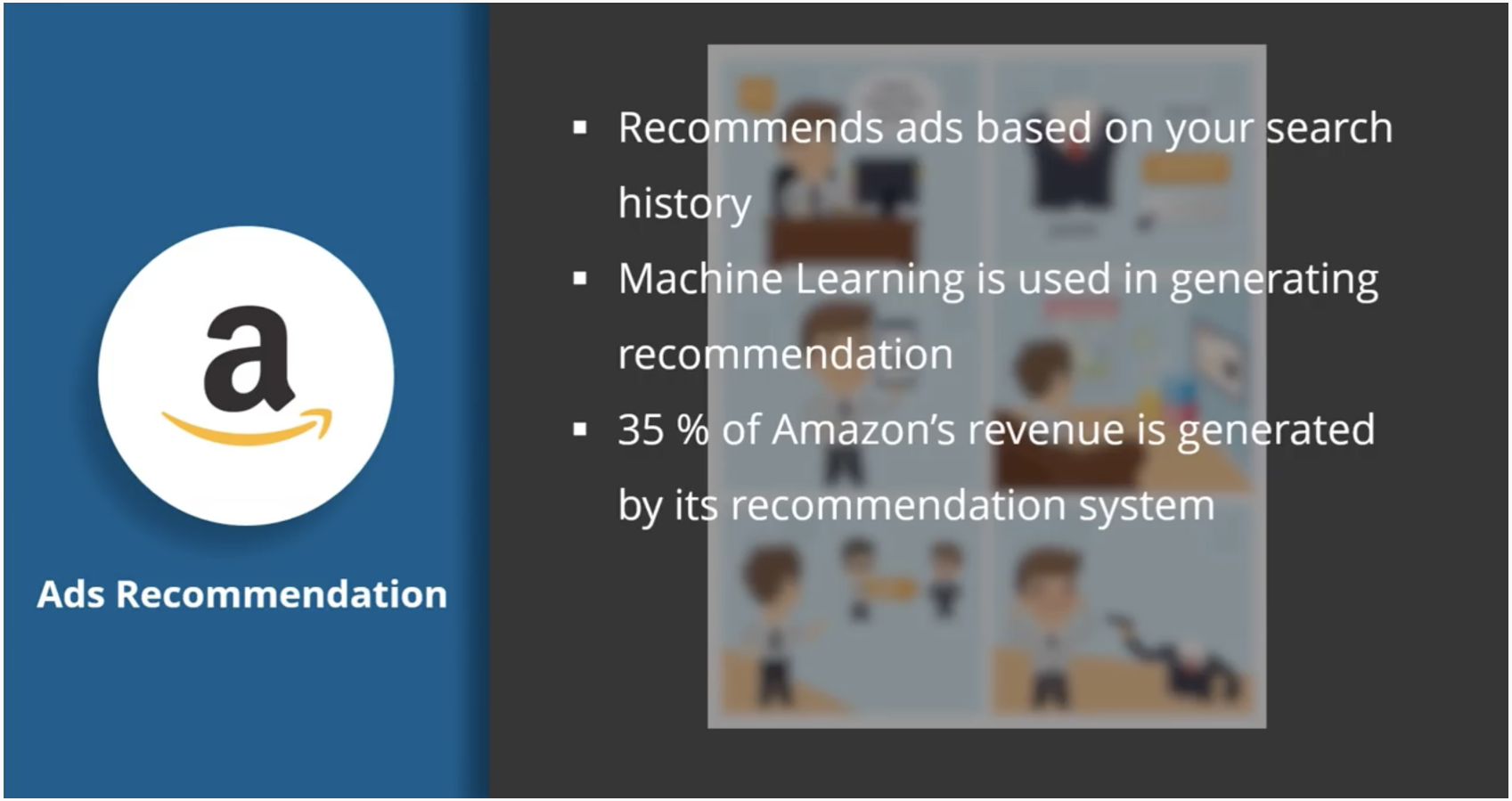
- Long-standing problem in AI. Hard to write a program to solve this
- We don't know how humans do it! Even if we did it would be an extremely complex program to write
- ML takes a data-driven approach. We collect lots of examples and train an algorithm with them

Machine Learning

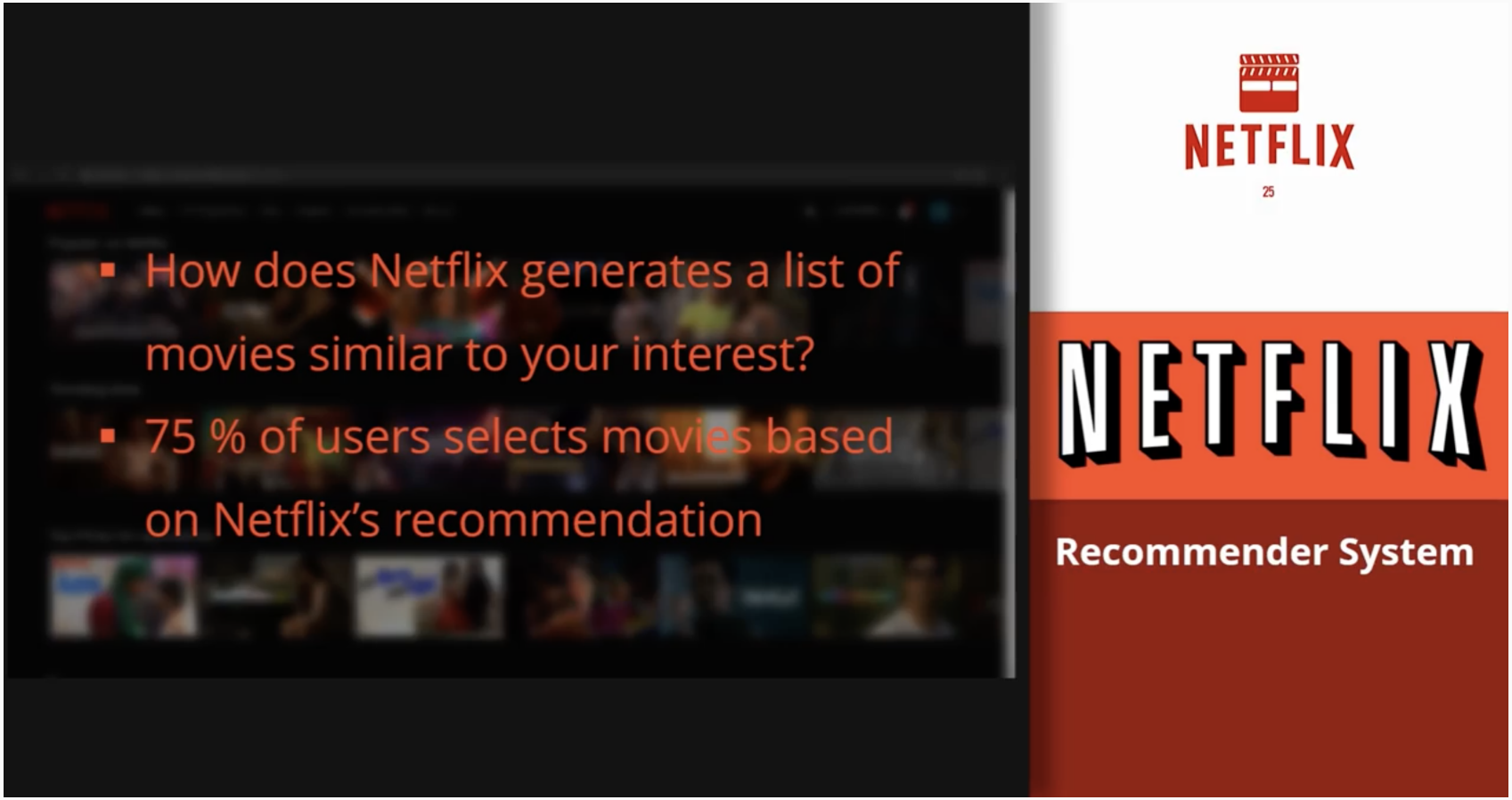
'A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.'

- Mitchell, T. (1997). Machine Learning. McGraw Hill. p. 2.

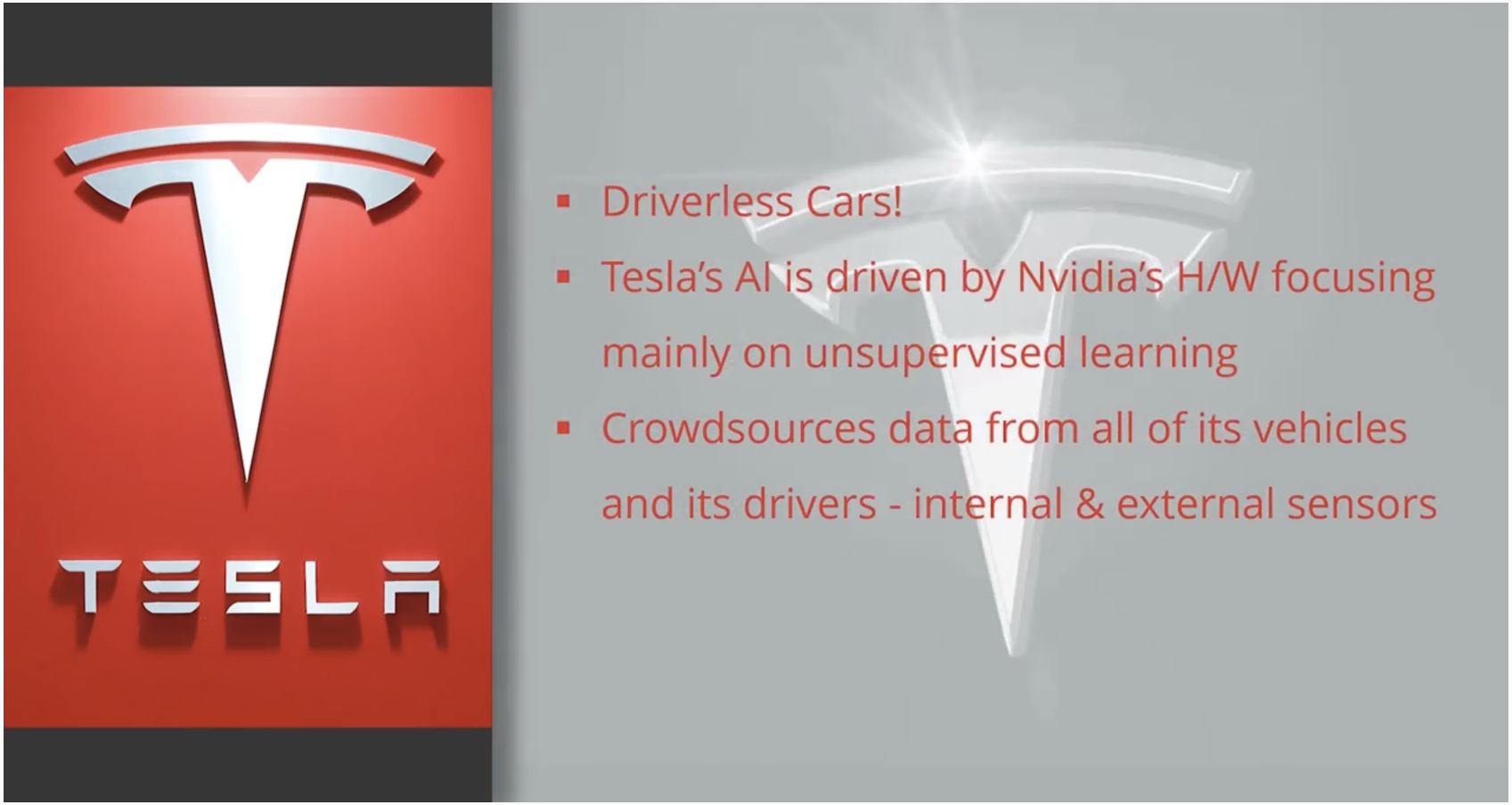
ML applications: recommendation



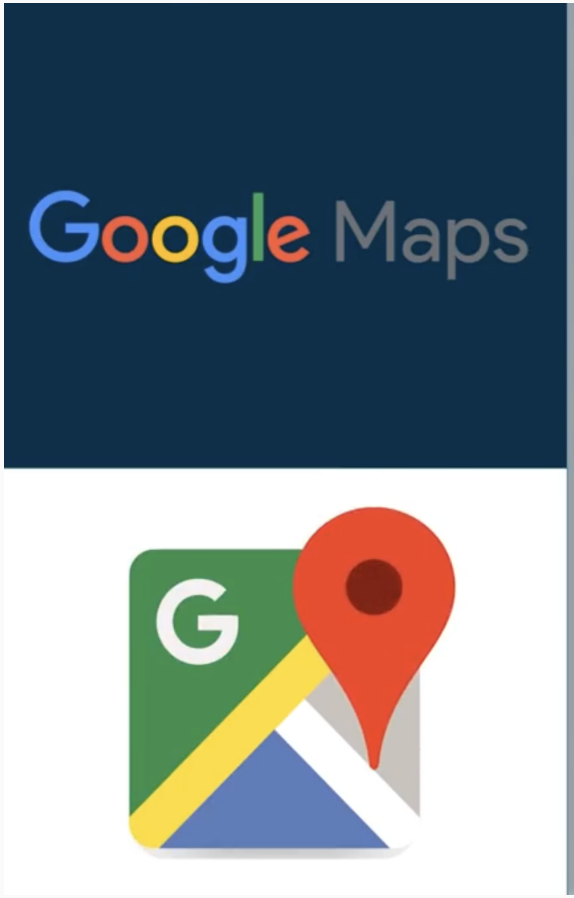
ML applications: recommendation



ML applications: self-driving cars



ML applications: navigation

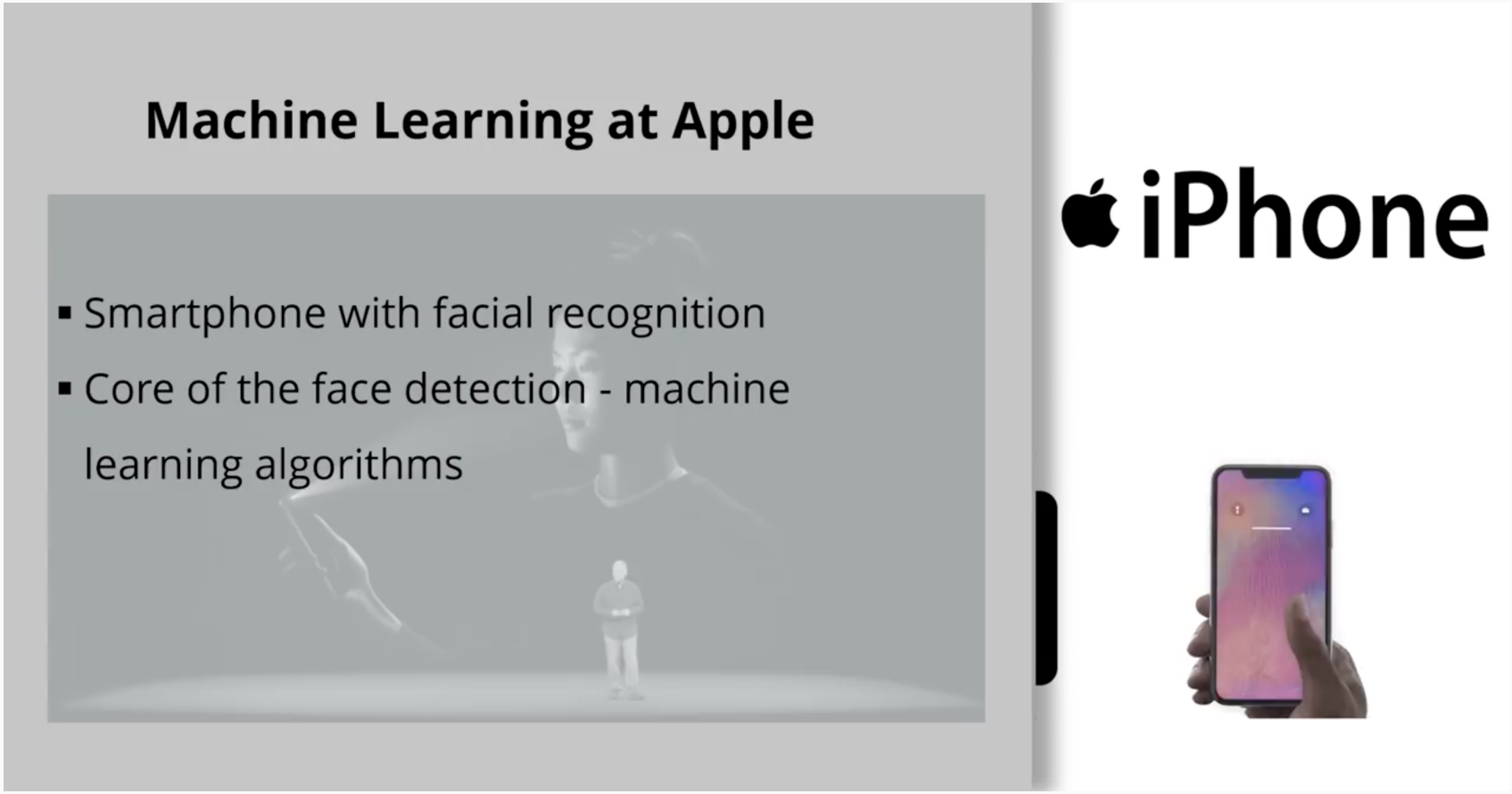


Faster Route Selection

- Google Maps - THE app which we use every time we go out
- Despite of the usual traffic, you are on the fastest route
- Everyone who is using the Google Maps is contributing in making the app more accurate

ML applications: navigation

ML applications: smart phones



Outline

Background

Artificial Intelligence, Machine Learning and Deep Learning

Example applications

Machine Learning: the settings

ML Tasks

Performance Measures & experience

Machine learning: important concepts

Generalisation, overfitting and underfitting

Model selection

Model example: k-nearest neighbours

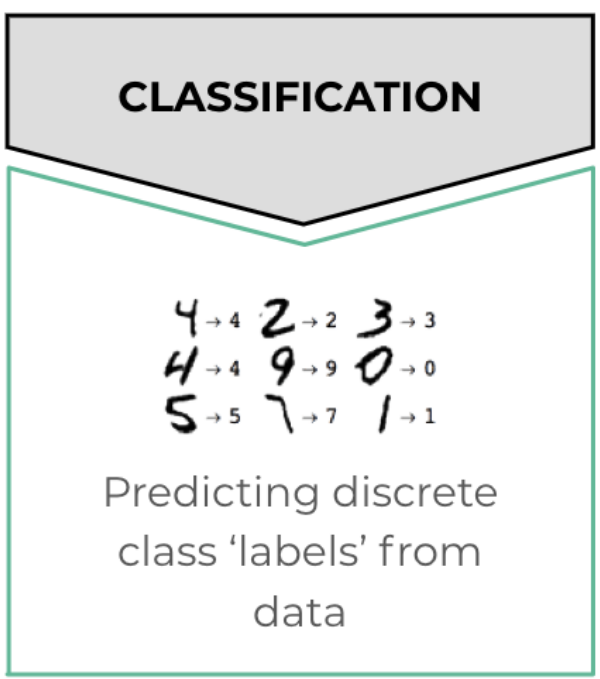
Machine Learning

Recall Mitchell's definition of a learning algorithm:

'A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.'

What kinds of tasks T are machine learning algorithms suited to?

Machine Learning - task T



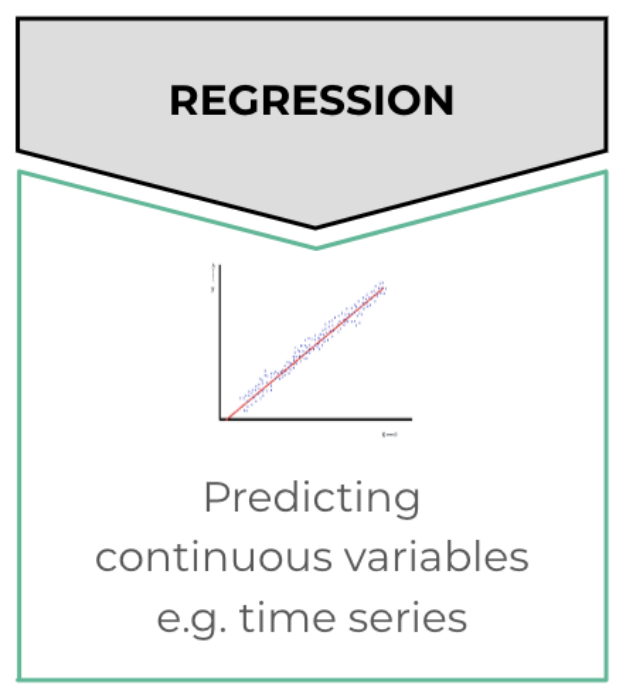
The data is generated according to the joint distribution $p(\mathbf{x}, \mathcal{C})$, where $\mathbf{x} \in \mathbb{R}^n$ and $\mathcal{C} \in \{1, \ldots, k\}$.

The aim is to learn a function $f : \mathbb{R}^n \to \{1, \ldots, k\}$ that classifies the data.

Typically split into a model of the conditional distribution $p(C|\mathbf{x})$ and some decision rule. This is a **discriminative** task.

Examples: object recognition, sentiment analysis.

Machine Learning - task T



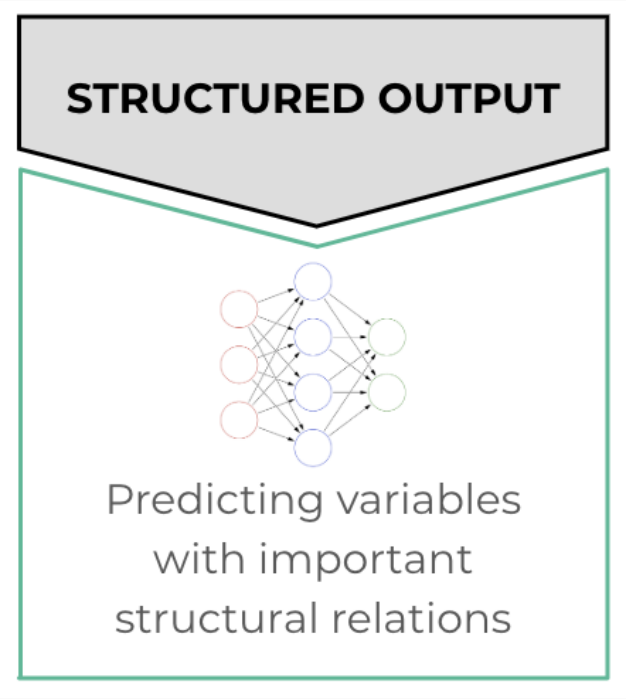
The data is generated according to the joint distribution $p(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}$. The data could be sequential in nature.

This task is similar to classification, but the output type of the model is different.

Could learn a deterministic mapping, or train a model to output the parameters of a distribution.

Examples: insurance claims, financial markets.

Machine Learning - task T



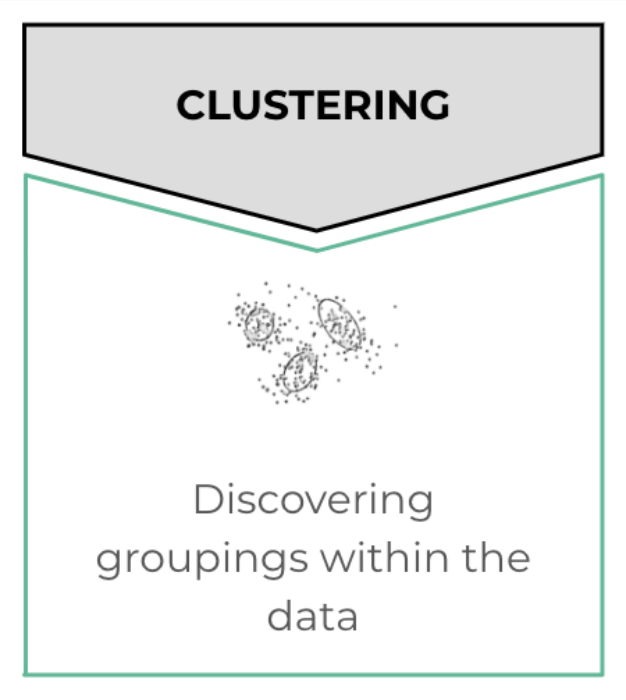
The data resides in a highly structured space, such as parse trees, graphs, or the space of valid sentences for a translation task.

Models typically output vector representations that encode structural objects.

As before, the model output could be deterministic or it could be a distribution over possible outputs.

Examples: OCR, transcription, language translation.

Machine Learning - task T



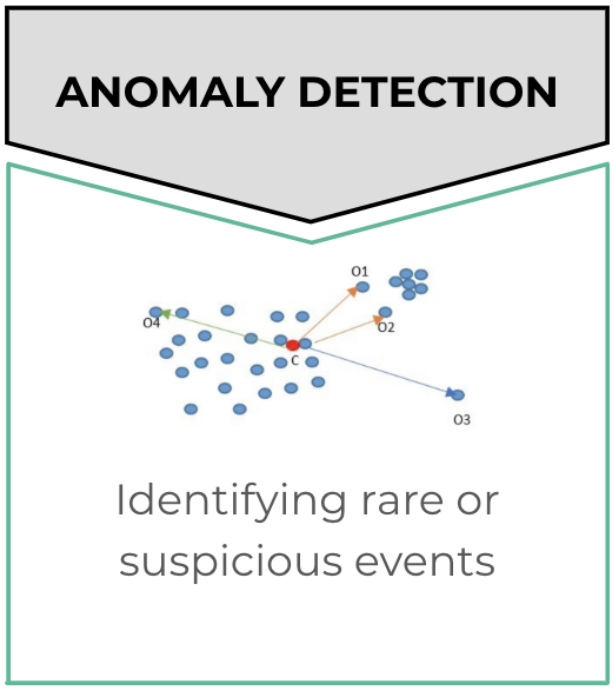
The data often resides in a high-dimensional space, but is concentrated in a relatively small number of regions.

This is a type of unsupervised learning, as there are no labels or outputs are part of the data..

The model output assigns each data point to a cluster, often the number of clusters being selected by the user.

Examples: market segmentation, social networks, recommender systems.

Machine Learning - task T



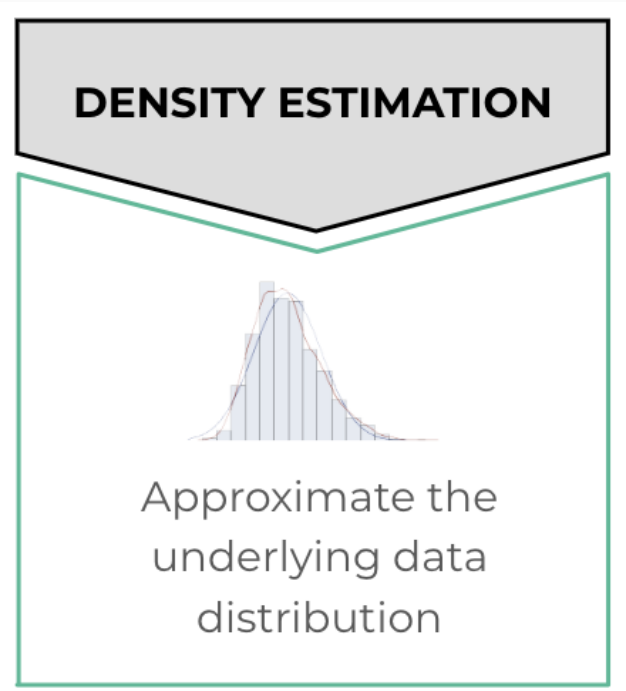
The data is generated according to an underlying distribution or process, with a few anomalies or outliers.

Anomaly detection can be supervised (if labels for example outliers exist) or unsupervised (no labels).

Outliers might be rare objects, or unexpected bursts of activity.

Examples: fraud detection, system health monitoring, network intrusion.

Machine Learning - task T



The data is generated according to $p(\mathbf{x}, \mathbf{y})$ or just $p(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}$.

The task is to learn the underlying (joint) probability density function that generated the data.

This **generative** modelling approach can be useful for other tasks, as well as generating new data examples similar to the training set.

Examples: speech synthesis, protein design.

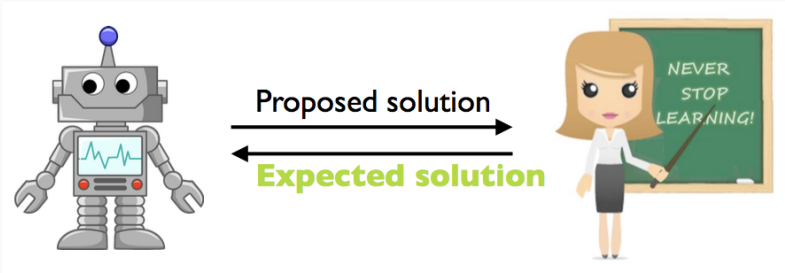
Machine Learning - performance P

- Need a quantitative measure to train and evaluate the model
- Choosing a performance measure is not always trivial
- Measure depends on task and model output
  - Classification accuracy
  - Least squares error
  - Cross-entropy loss
  - Discounted sum of rewards
  - Mean average precision
  - ...
- Typically evaluate model performance on a held-out test set
- In addition, a separate validation set may be used

Machine Learning - experience E

Supervised and unsupervised learning are two broad categorisations of the type of experience, or data, that models are provided with.

- **Supervised** learning algorithms are provided with a training set of data features and associated labels

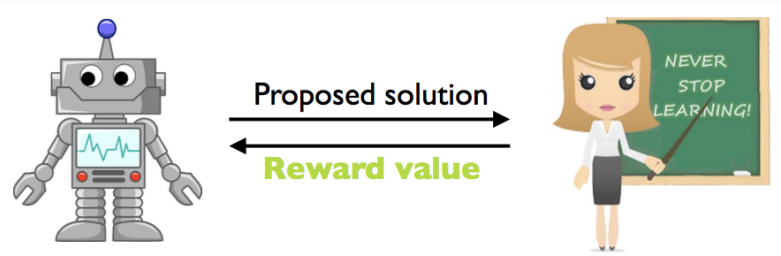


- **Unsupervised** learning algorithms are provided with a training set of data features, and are tasked with finding useful representations or structure in the data

Machine Learning - experience E

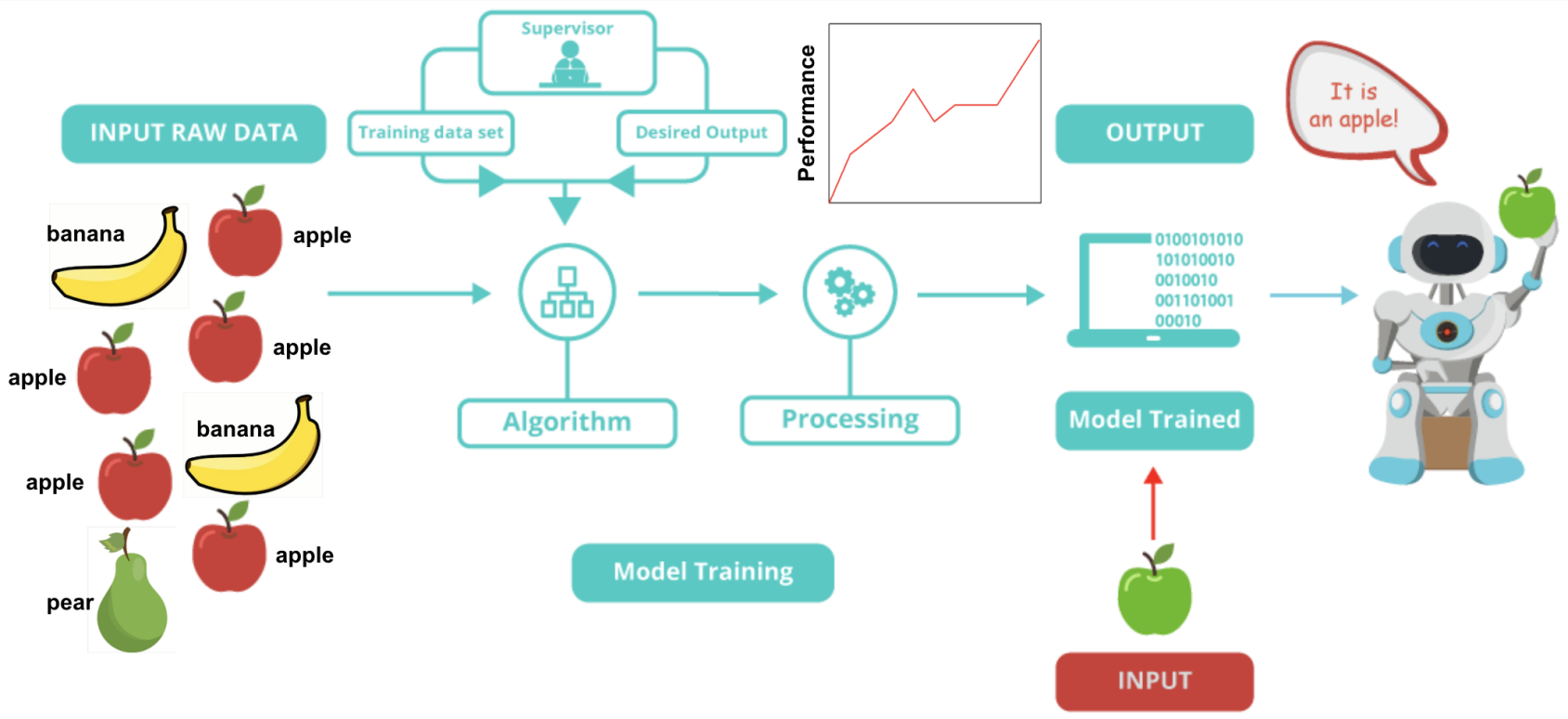
Some machine learning algorithms are provided with an interactive experience with which to learn from.

Reinforcement learning algorithms interact with an environment, and are provided with features of the environment, as well as a reward signal. The models must use this feedback from the environment to learn the optimal way to interact with the environment.



Machine Learning definition revisited

'A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.'



Outline

Background

- Artificial Intelligence, Machine Learning and Deep Learning
- Example applications

Machine Learning: the settings

- ML Tasks
- Performance Measures & experience

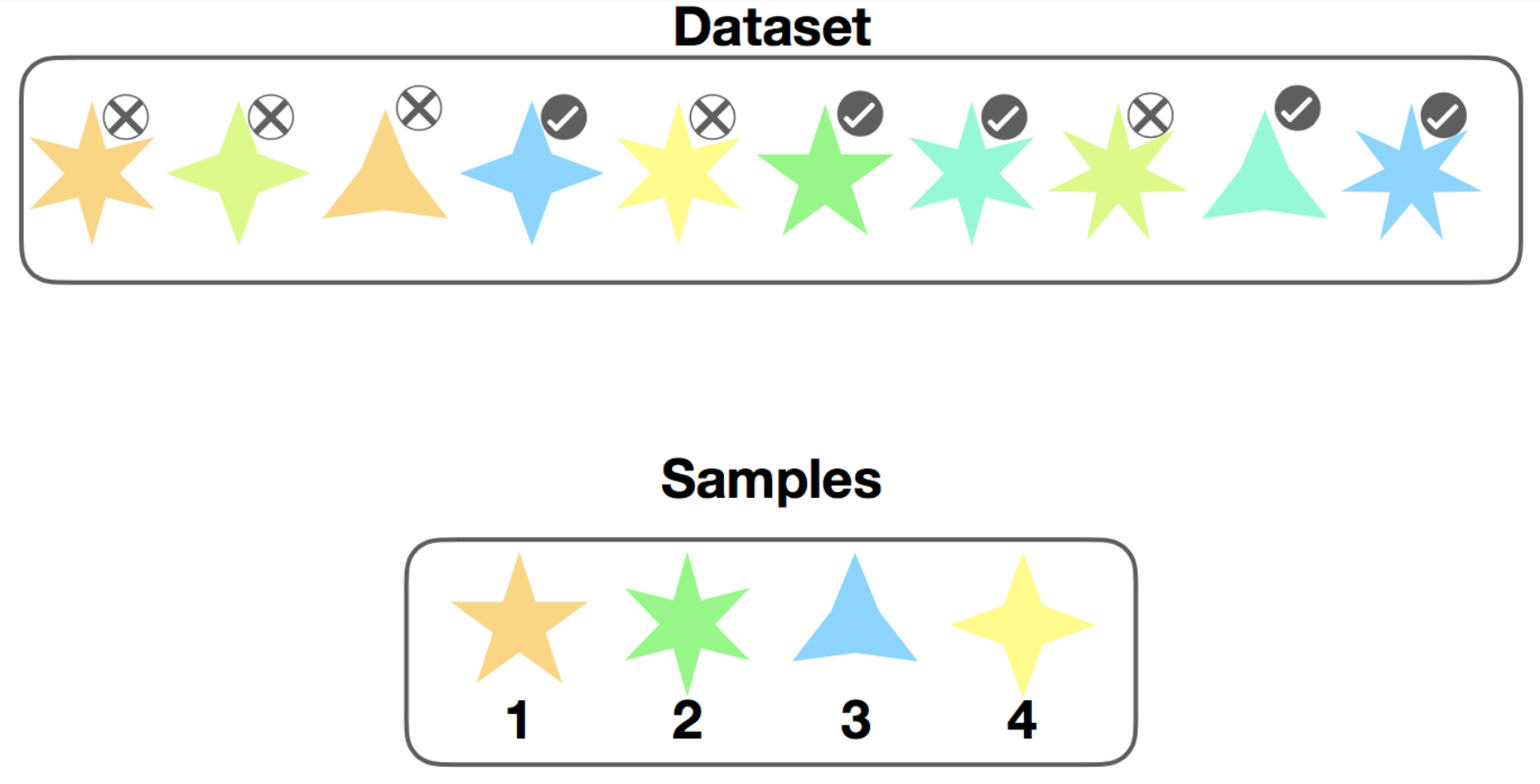
Machine learning: important concepts

- Generalisation, overfitting and underfitting
- Model selection

Model example: k-nearest neighbours

An ML challenge

- A binary classification task
- Can you predict the right labels for our test samples?

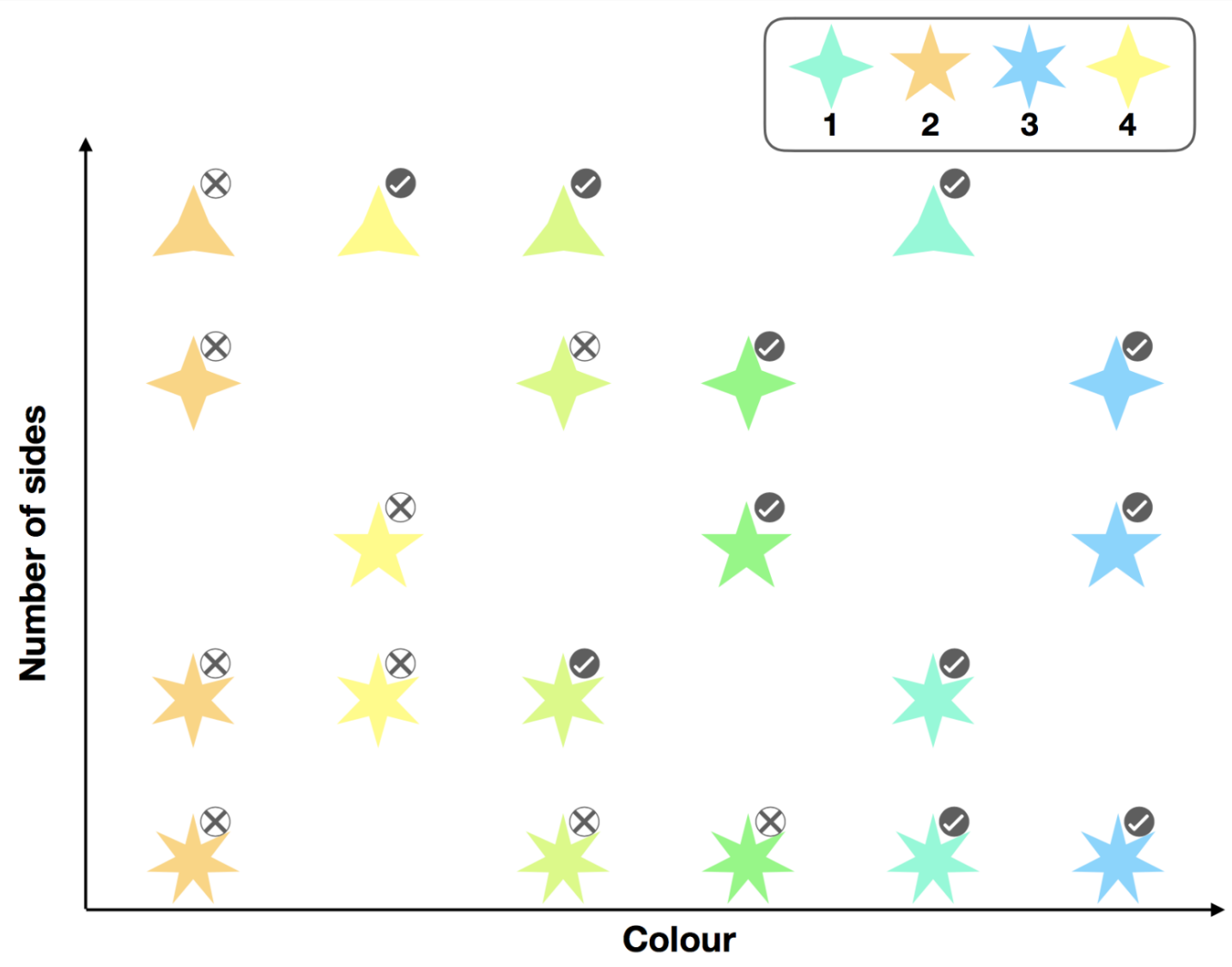


An ML challenge

- ...need more data?
- What are the relevant features?

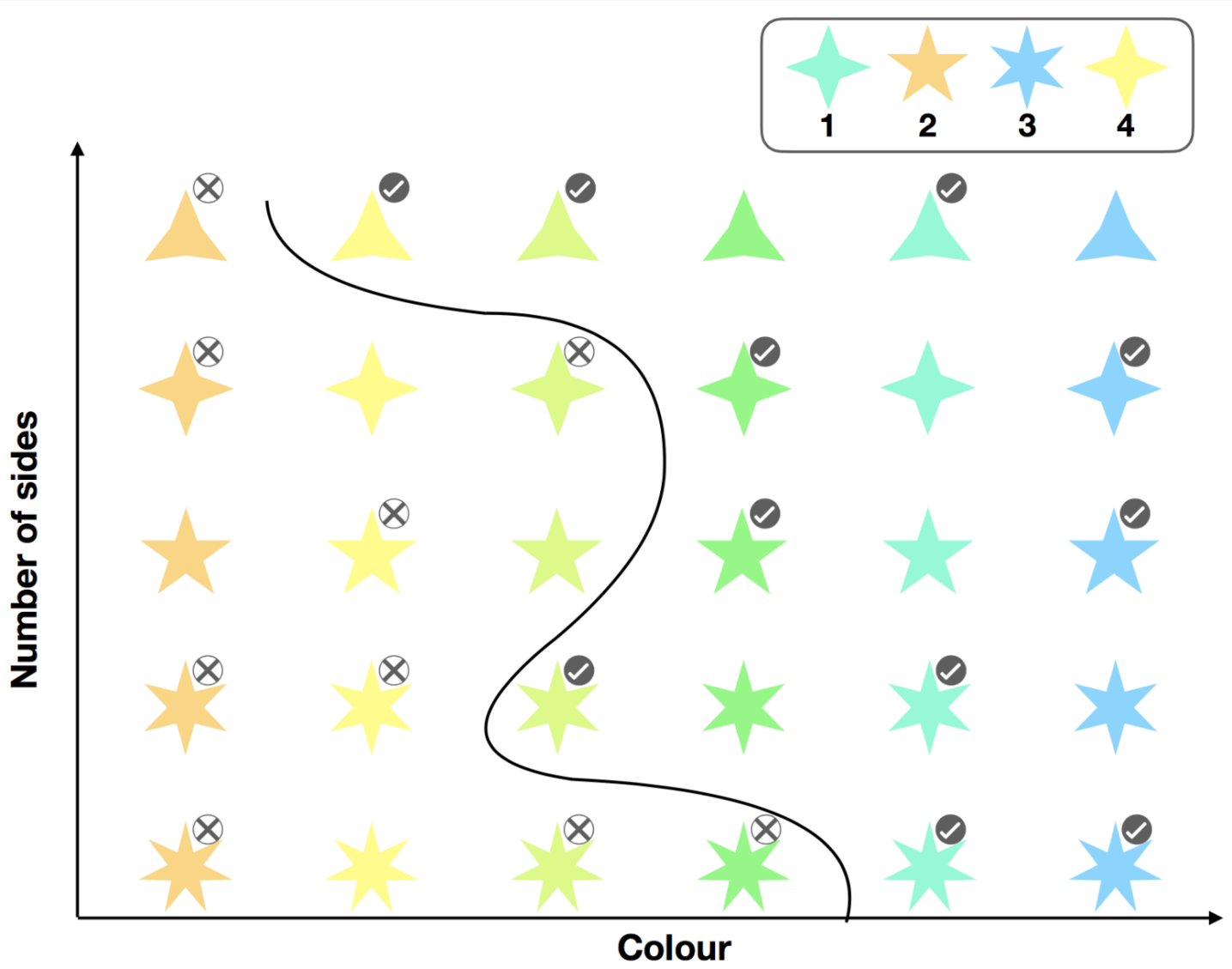
An ML challenge

- Can you predict the labels now for the test samples?



An ML challenge

- Can you predict the labels now for the test samples?



An ML challenge

- Large amounts of data are needed to make accurate predictions
- There will often be additional data features that may or may not be relevant for the task
- Selection of the right features and data representation is essential
- A good machine learning model will learn from the available data and use the learned knowledge to apply to test samples to make predictions

Machine Learning - generalization

The ability to perform well on previously unseen inputs is **generalization**.

- Generalization is usually measured on a test set (separate to training set)
- Underlying assumption is that training set and test set are i.i.d.
- Parameters of the model are adjusted according to the training set
- Model **overfits** if it performs poorly on the test set
- Model **underfits** if the training error is too large

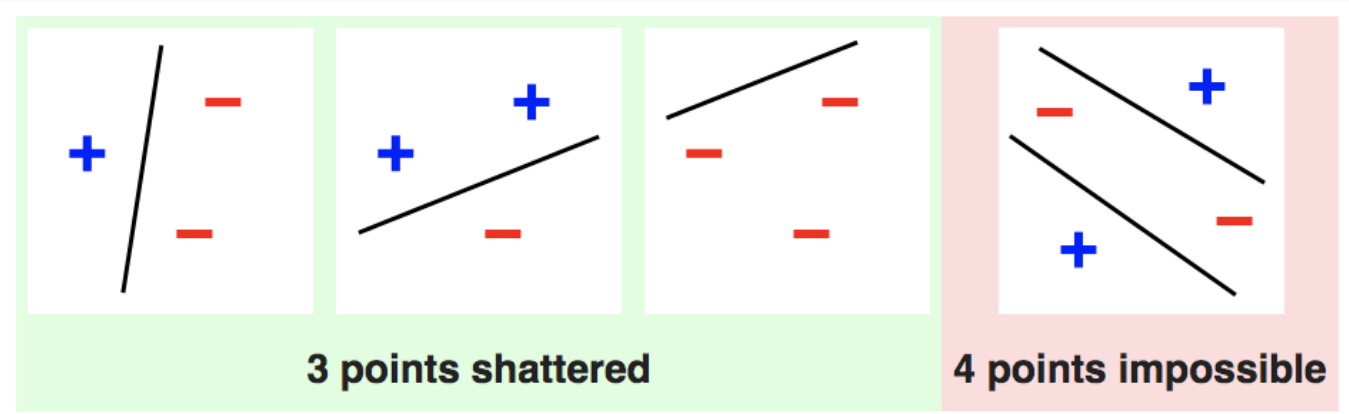
Overfitting/underfitting can be controlled through the model complexity and regularisation.

Model capacity

- A model's capacity is informally defined as its ability to fit a wide variety of functions (for regression, classification or any other task)
- Capacity is controlled by model **hyperparameters**:
  - For instance, the number and width of layers in a neural network
  - Polynomial degree in regression problems
- Models with low capacity will struggle to make good predictions for a complex dataset
- Models with high capacity could have the ability to simply memorise the dataset
- The set of functions that the model can choose from is called the **hypothesis space**

VC dimension

- The **Vapnik-Chervonenkis dimension**, or VC dimension is a measure to quantify model capacity from statistical learning theory
- The VC dimension measures the capacity of a binary classifier
- It is defined as being the largest possible value of m for which there exists a training set of m different data points that the classifier can label arbitrarily



Regularisation

- In practice, a common method for preventing overfitting is **regularisation**
- In general, we identify capacity with the size of the hypothesis space
- In addition, we can prefer some solutions within the hypothesis space over others. In particular, we should prefer simpler solutions over complex ones if they fit the data well (Occam's razor)
- Regularisation provides a means of doing just that; it is roughly defined as any modification we make to a learning algorithm that is intended to reduce its generalisation error but not its training error

Model validation

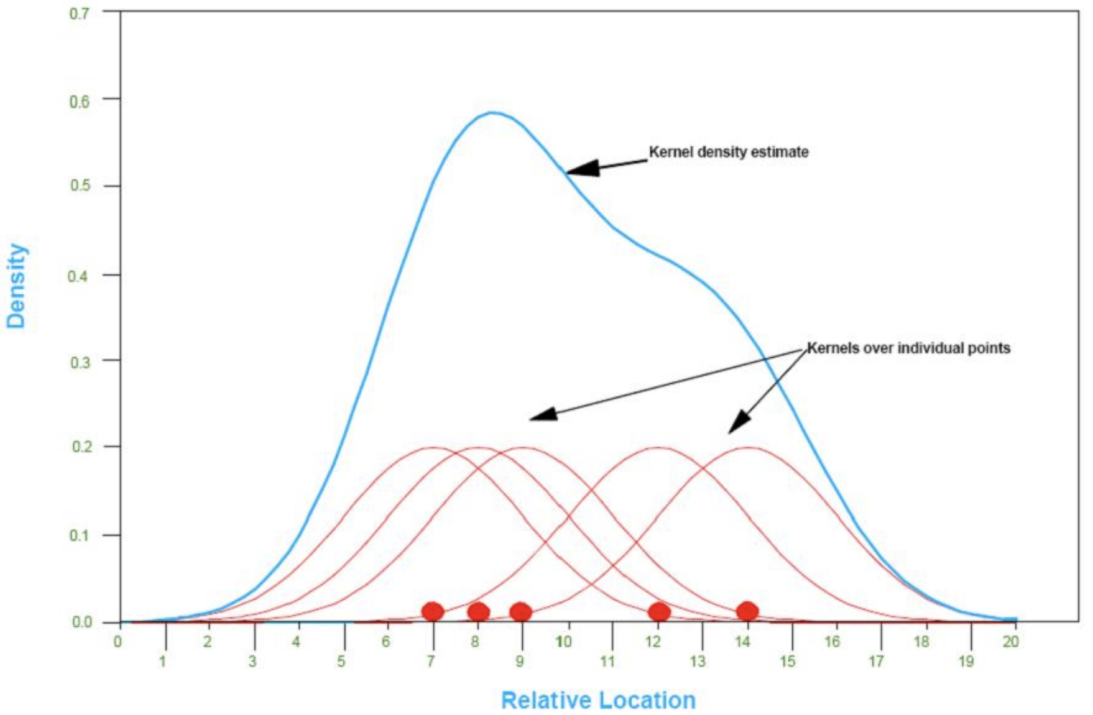
- Recall that the goal of fitting a machine learning model is to generalise
- Good generalisation is achieved by avoiding overfitting and underfitting
- Model capacity should be chosen appropriately - how do we choose hyperparameters?
- A typical process is to fit a range of models for different hyperparameter choices, evaluate them on a **validation set**, and choose the best one
- Finally, there is often an additional **test set** that is used to give a final unbiased evaluation of the model
- Our available data is split into these three partitions, for example as a 50/25/25 split

Cross validation

- Cross validation is a common procedure used for model selection
- The procedure works by iterating the training—validation process for different partitions of the available data
- For example, in k-fold cross validation, the data is partitioned into k equal-sized subsets
- On each iteration, a single partition is chosen as the validation set and the remaining $k - 1$ partitions are used for training
- This is repeated k times, once for each partition used for validation
- The results are then averaged to provide a single estimate
- In *stratified k-fold cross validation* we ensure the proportion of classes within each partition is roughly the same

Nonparametric models

- Nonparametric models can be thought of as models with potentially infinite capacity
- The capacity of the models grow with the size of the dataset
- Examples of nonparametric models include k-nearest neighbours, kernel density estimation, support vector machines and Gaussian processes



Outline

Background

Artificial Intelligence, Machine Learning and Deep Learning

Example applications

Machine Learning: the settings

ML Tasks

Performance Measures & experience

Machine learning: important concepts

Generalisation, overfitting and underfitting

Model selection

Model example: k-nearest neighbours

Model example: k-nearest neighbours

- k-nearest neighbours (k-NN) is a simple and intuitive example of a machine learning model that can be used for both classification and regression problems
- It is a nonparametric model
- It is used for supervised learning
- It is an example of a *lazy learner*; it defers construction of a target function until a test query is made (as opposed to an *eager learner*, which builds an explicit target function during training that is applied to test cases)
- Predictions are made for a query example by looking at the labels for its k nearest neighbours

Model example: k-nearest neighbours

- k-NN binary classification takes the k nearest neighbours of a query instance and assigns the majority class
- The Euclidean distance is usually used to determine the nearest neighbours of a query instance, but ℓ_1 or ℓ_∞ norms are sometimes used instead
- k is a hyperparameter of the algorithm and can be chosen with validation techniques
  - Small k gives good resolution but high sensitivity to noise (risk of overfitting)
  - Large k gives a low resolution but is more robust to noise (risk of underfitting)
- For a large dataset the algorithm can be slow
- Refinements to k-NN can be made where weightings are applied according to distance from the query example

Model example: k-nearest neighbours

We look at a simple binary classification example to predict customer's T-shirt sizes from their height and weight:

Height (cm)	Weight (kg)	Size
158	58	M
158	59	M
158	63	M
160	59	M
160	60	M
163	60	M
163	61	M
160	64	L
163	64	L
165	61	L
165	62	L
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L
170	68	L

Model example: k-nearest neighbours

- In our implementation of the algorithm, we will use the ℓ_2 distance to compute nearest neighbours

$$d(x, y) = \left(\sum_{i=1}^{m} (x_i - y_i)^2 \right)^{\frac{1}{2}},$$

  where m is the number of features (2 in our case)
- We will choose $k = 5$ nearest neighbours to make predictions
- Suppose we have a test query that is a customer with height **161cm** and weight **61kg**

Model example: k-nearest neighbours

We need to compute distances from this test example to all examples in the dataset:

Height (cm)	Weight (kg)	Size	Distance
158	58	M	4.2
158	59	M	3.6
158	63	M	3.6
160	59	M	2.2
160	60	M	1.4
163	60	M	2.2
163	61	M	2.0
160	64	L	3.2
163	64	L	3.6
165	61	L	4.0
165	62	L	4.1
165	65	L	5.7
168	62	L	7.1
168	63	L	7.3
168	66	L	8.6
170	63	L	9.2
170	64	L	9.5
170	68	L	11.4

Model example: k-nearest neighbours

Find the $k = 5$ nearest neighbours:

Height (cm)	Weight (kg)	Size	Distance	
158	58	M	4.2	
158	59	M	3.6	
158	63	M	3.6	
160	59	M	2.2	**3**
160	60	M	1.4	**1**
163	60	M	2.2	**3**
163	61	M	2.0	**2**
160	64	L	3.2	**5**
163	64	L	3.6	
165	61	L	4.0	
165	62	L	4.1	
165	65	L	5.7	
168	62	L	7.1	
168	63	L	7.3	
168	66	L	8.6	
170	63	L	9.2	
170	64	L	9.5	
170	68	L	11.4	

The majority class is M!

Model example: k-nearest neighbours

- Typically we would normalise the features before running the k-NN algorithm
- Features that are measured in different units (or otherwise vary on different scales) will have a greater or lesser impact on the distance calculation
- Features are often 'whitened' to ensure they are on the same scale:

$$\hat{x} = \frac{x - \overline{x}}{\sigma},$$

where $\overline{x}$ denotes the mean feature vector and σ is the standard deviation, computed over all data examples