Machine Learning CSE2510 – Lecture 1.1

Gosia Migut
Odette Scharenborg
David Tax
Amira Elnouty

TU Delft

Welcome to week 1 - lecture 1

- Course overview / Administrative info
  - (see also Brightspace for all information, slides, assignments, reading material, etc)
- Machine Learning (ML): introduction
- The ML pipeline
- Measurements, features, objects, datasets

TU Delft

This course (5 ects)

- **Goal:** acquaint students with the basic Machine Learning concepts and algorithms
- Specifically:
  - parametric and non-parametric density estimation
  - linear and non-linear classification
  - unsupervised learning
  - performance evaluation of predictive algorithms
  - ethical issues in machine learning

The learning objectives

After this course, you are able to:

- Explain the basic concepts and algorithms of machine learning and underlying statistical concepts
- Implement and apply ML algorithms in Python
- Explain the concept of and identify (implicit) bias in data and ML algorithms

Teaching staff

	Role
Gosia Migut	Course coordinator + responsible lecturer
Odette Scharenborg	Responsible lecturer
David Tax	Co-lecturer
Amira Elnouty	Lab coordinator
Jordi Smit	Head TA

TU Delft

Time distribution of the 5 ECTS

Week	1	2	3	4	5	6	7	8	9	10	Total
Attend lectures	4	4	4	4	4	4	4	4			32
Reading	4	4	4	4	4	4	4	4			32
Lab sessions	4	4	4	4	4	4	4	4			32
Assignments	3	3	3	3	3	3	3	3			24
Prepare exam									15		15
Do exam										3	3
Total	15	15	15	15	15	15	15	15	15	3	138

5 ECTS = 140 h

TU Delft

Course structure

- 2 lectures each week (Tue & Fri (Thu!))
- 2 shared labs (Tue & Thu)
  - 4 hours expected
  - Voluntary
  - With TA support (not today)
  - Topics are directly related to the lecture material

Final grade

- Digital exam only:
  - Open questions
  - Programming questions
  - Multiple choice questions
- Resit in Q2
- To prepare for the digital exam:
  - Do lab assignments
  - Read material
  - Attend lectures
  - Participate in the exercises during lectures
  - Do the practice mid-term/final exams

Communication

- Content-based questions:
  - Talk to us during lecture breaks or after the lecture
  - Ask your fellow student (in person, Mattermost:

  https://mattermost.ewi.tudelft.nl/signup_user_complete/?id=esjkmbhpcbyn9qmy954z1wkgkw)

  Note: The teaching staff will not answer any questions on Mattermost
  - Ask the TAs during labs

  Note: Content-based questions via e-mail will not be answered

TU Delft

Communication (2)

- Admin questions:
  - During the first 5 minutes of the lecture
  - E-mail ml-cs-ewi@tudelft.nl
  - Note: Email to our personal mailboxes will not be answered
  - Please use a friendly header to start your e-mail (Dear Gosia/Odette/David/Amira)

Course layout

Week	Topic	Lecturer
1	Course overview & introduction to ML	Odette Scharenborg
2	Parametric density estimation	David Tax
3	Non-parametric density estimation	Gosia Migut
4	Linear classification	Gosia Migut
5	Responsible machine learning	Odette Scharenborg
6	Non-linear classification	Odette Scharenborg
7	Unsupervised learning	Gosia Migut
8	Evaluation & Q&A	David Tax

This week's lab

- Installing python
- If needed, do the python and numpy tutorials
- Includes many exercises
- Questions → ask the TAs on Thursday

Reading material + notations

- Reading material will come from different books (indicated on Brightspace)
- Different mathematical notations used

➔ A good practice for ‘real life’

- Notations in slides are to be used in this course

Questions?

TU Delft

Introduction to Machine Learning

TU Delft

Today's learning objectives

After practicing with the concepts of this week you are able to:

- Explain the basic ideas of machine learning and why and when it can be used
- Explain the machine learning pipeline from data to training to testing to evaluation

Q: What is machine learning?

- ML aims to identify regularities in the world (or data)

➔ Learning and generalisation

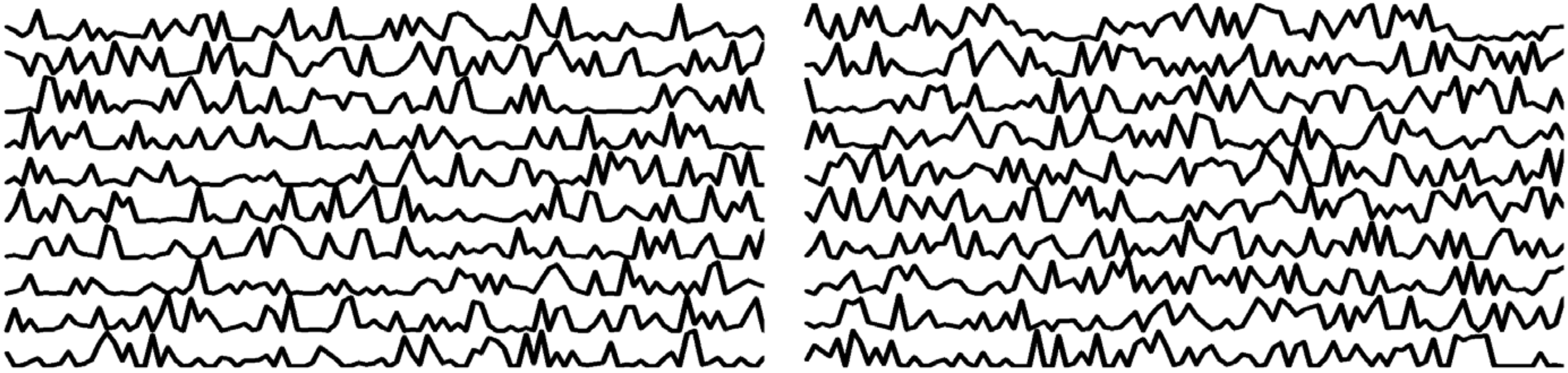
- So, we want to learn (from the world or data) and say something about a new situation

  == generalisation

- Learning == training on data

Why do we want to automate learning?

- 20 signals: from 2 different types / classes



Q: What is generalisation?

- Coming to general conclusions
  from (a limited number of) specific observations

(Something your parents probably told you **not** to do)

The Linda Problem

Linda is 31 years old, single, outspoken, very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations.

- Q : which of following alternatives is more probable?
  1. Linda is a bank teller
  2. Linda is a bank teller and active in the feminist movement

Q: A random person in the street

- What would you think?
  - Will the person be a professor?
  - Will the person be male?

- Possibility to make decisions using prior knowledge

➔ How do you obtain this prior knowledge?

Prior knowledge comes from measurements

Example Q: Can we predict gender from age?

- Measured data

	age > 85	age < 85
male	36	4965
female	106	4893

- Learning through counting

Slide adapted from Dr. Marco Loog

Predicting through counting

- Learn and predict based on a priori outcomes
  - Check (historical) data for expected outcomes
  - Assign to most likely, i.e., most occurring, outcome

➔ So, machine learning is about *probabilities*

Continuous measurements?

Rather artificial example:

- Observed:
  - 3 Dutch guys all being 19 decimeters
  - 3 German guys of 18, 19, and 20 dm

- New guy of 19 dm arrives
- Q: What nationality is he?

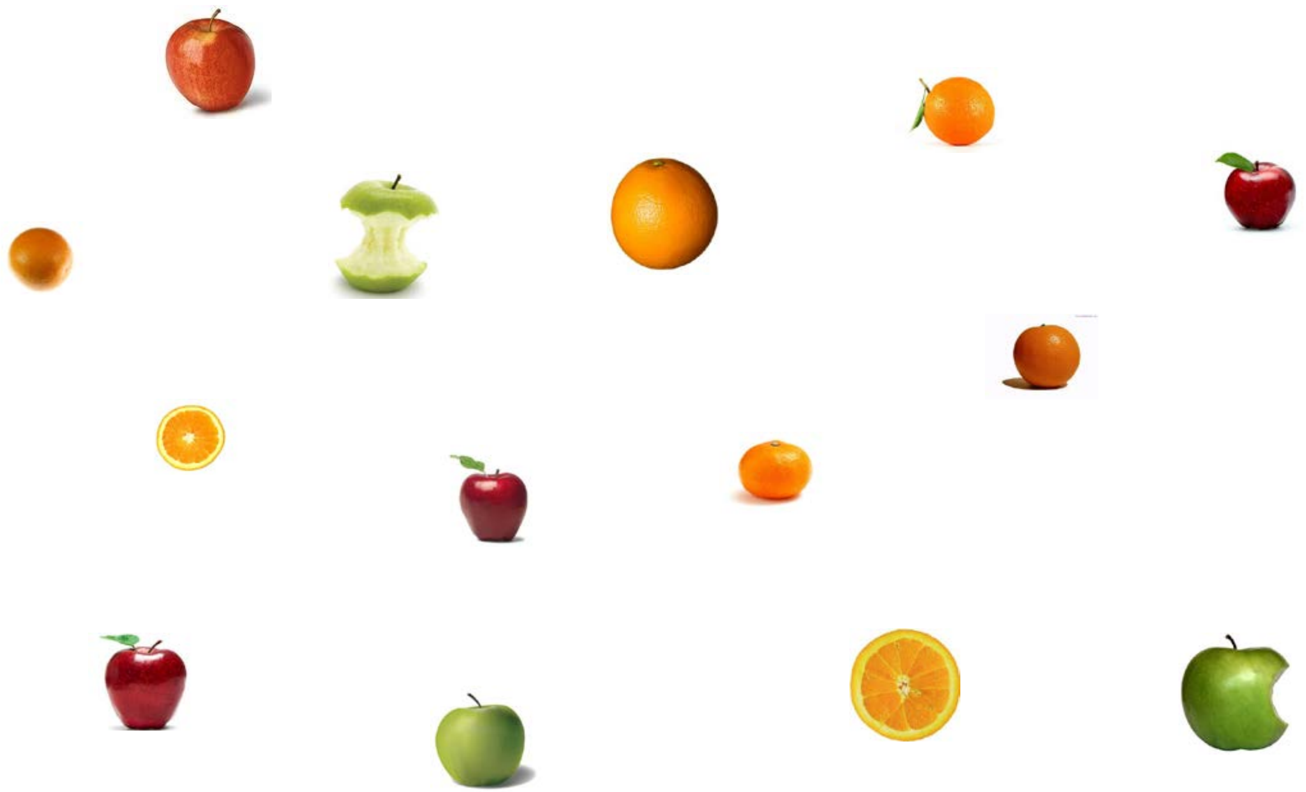
Continuous measurements?

- Say our measurement apparatus has improved

- So we get more accurate measurements… :
  - 3 Dutch guys : 19.267, 19.157, 18.812 decimeters
  - 3 German guys : 18.394, 18.771, 20.260 decimeters

  - New guy of 18.675 dm arrives.
  - Q: What nationality is he?

How do we go from observations to predictions?



Slide adapted from Dr. Marco Loog

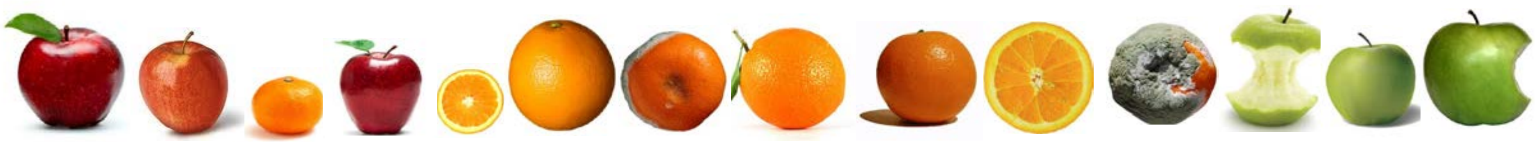
Supervised Learning

= **Learning by example**

- Given input-output examples, determine input-output function
- Function should be able to **generalise** to new and previously unseen examples

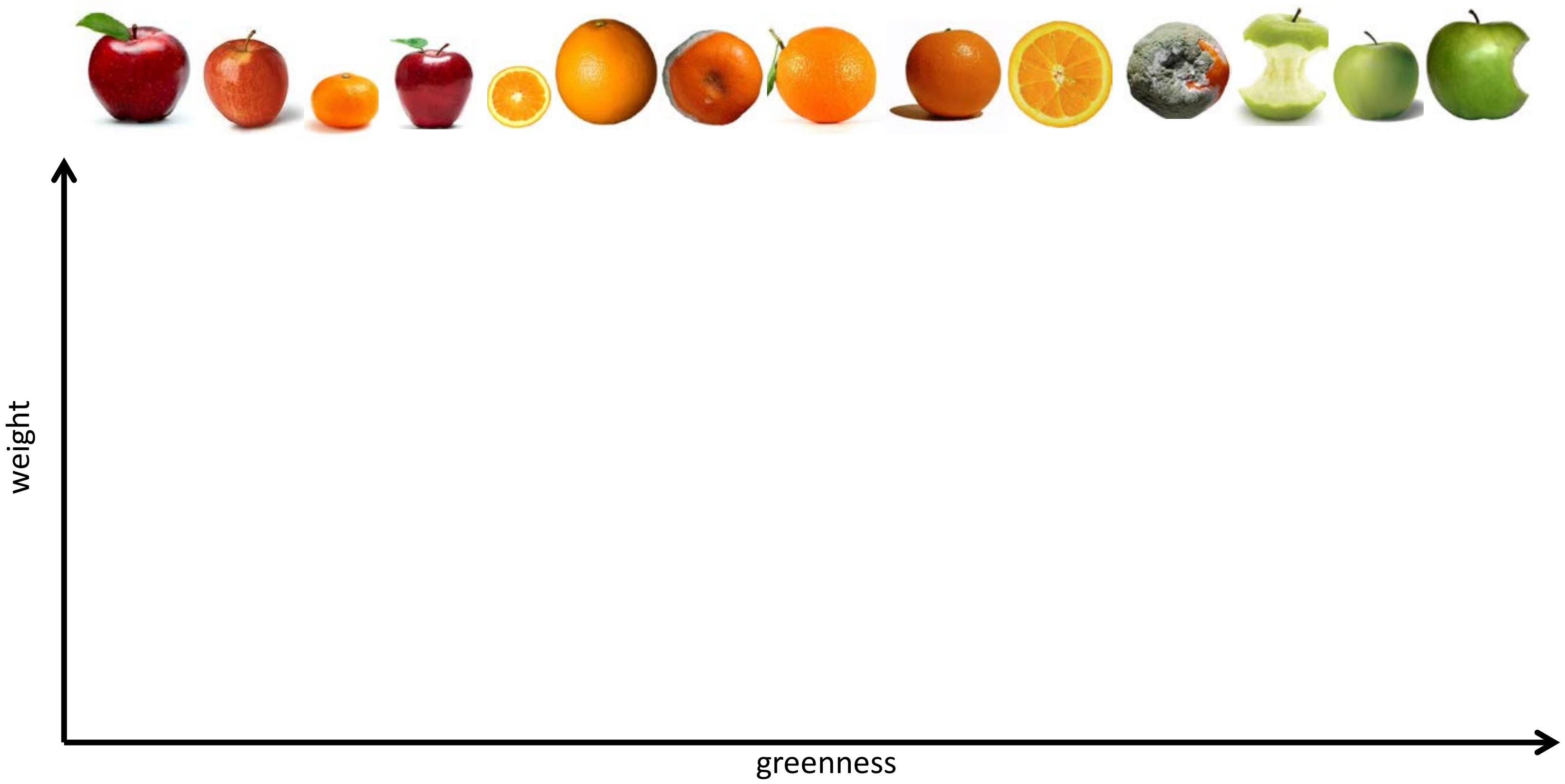
Slide adapted from Dr. Marco Loog

How does this work?

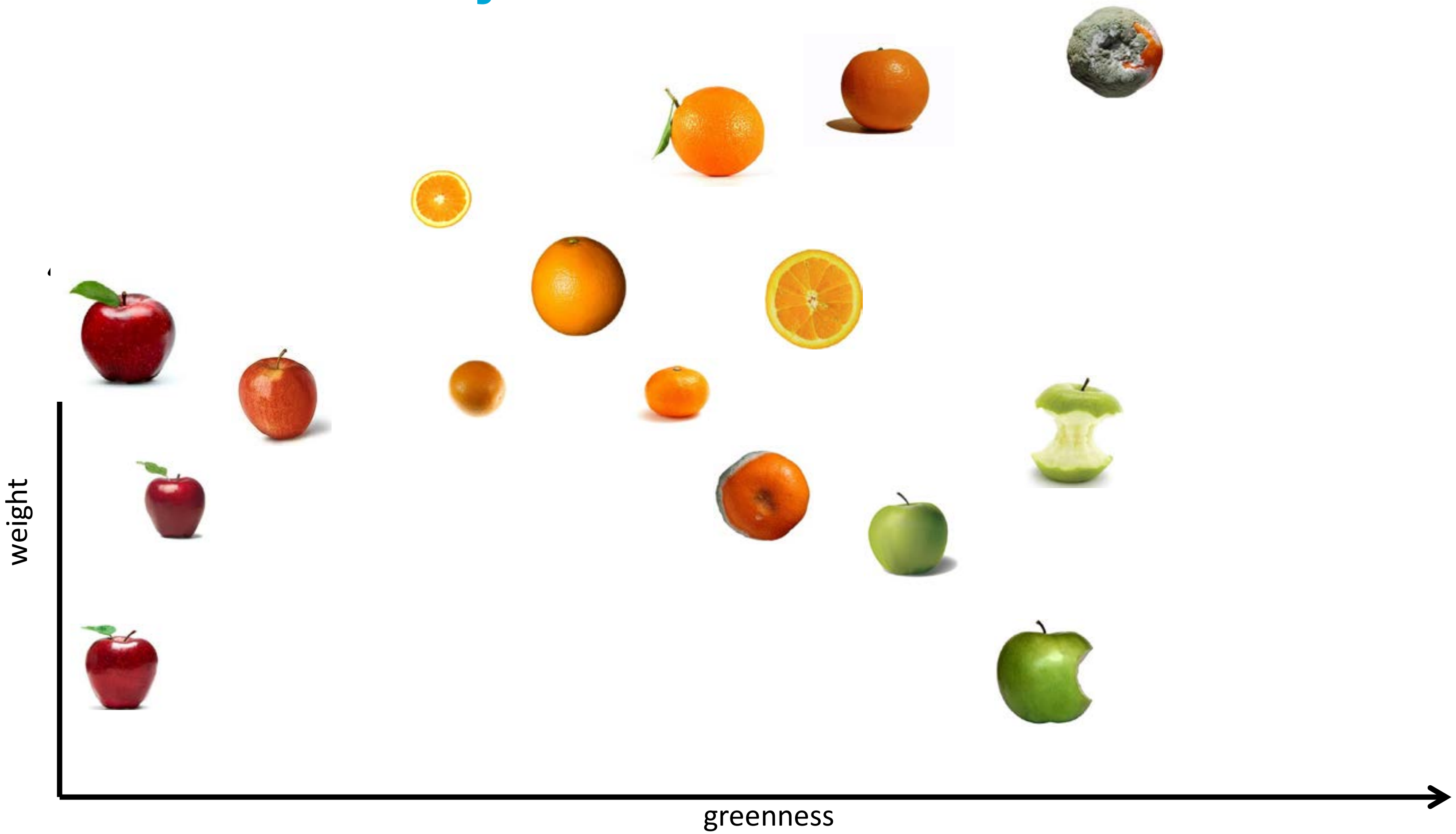


➔ Find a function that is able to split the apples and oranges into separate groups

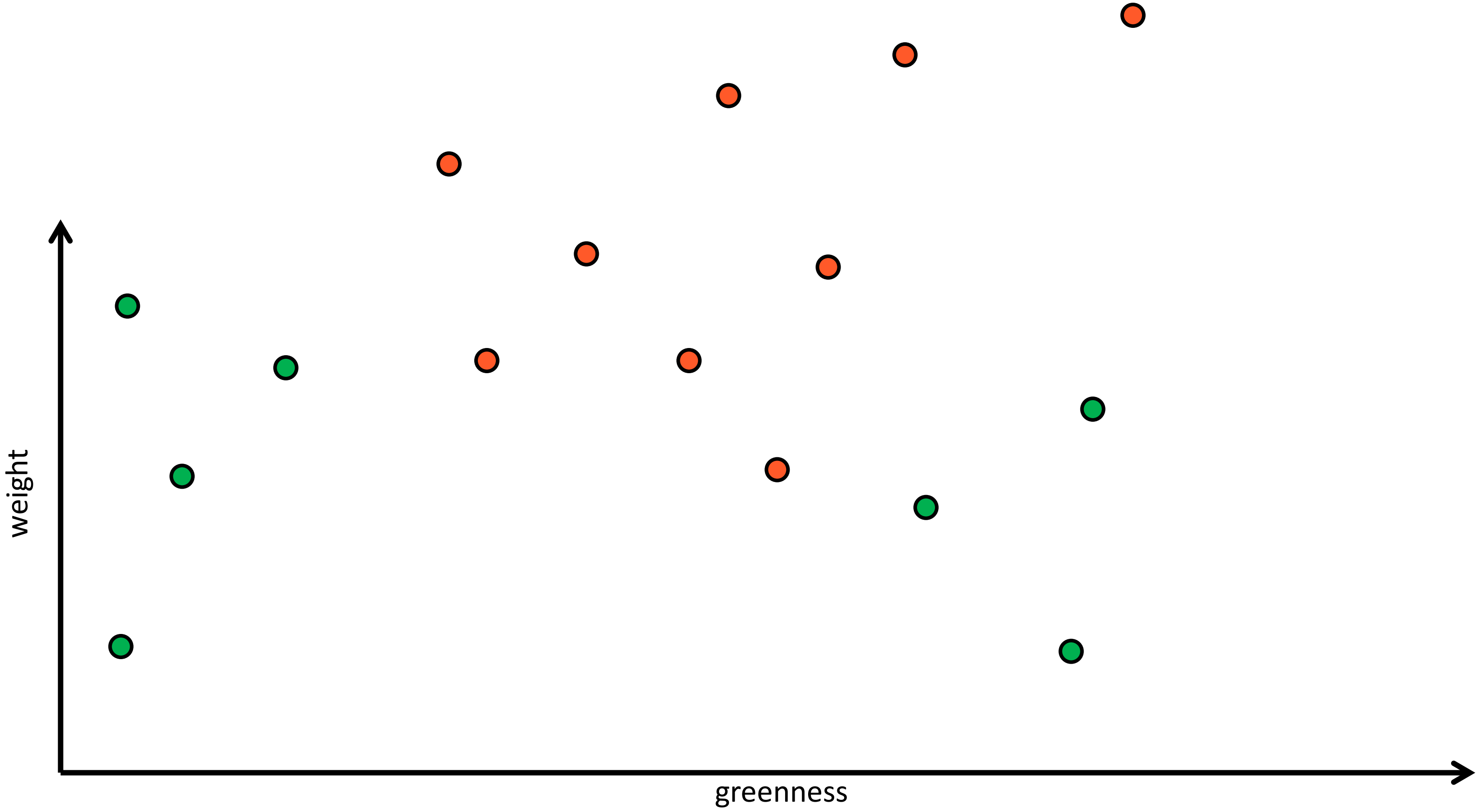
Take measurements



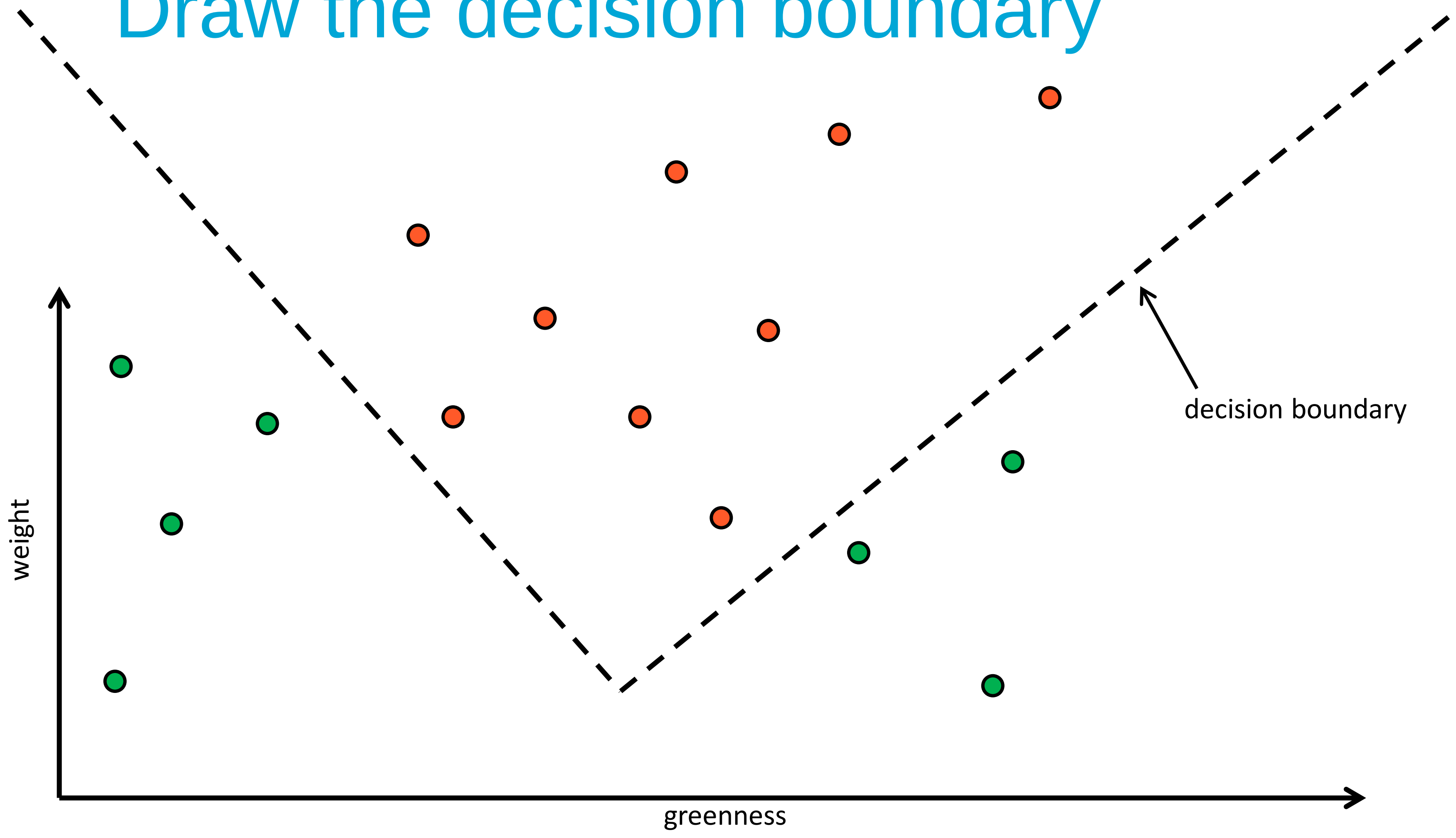
Plot each object



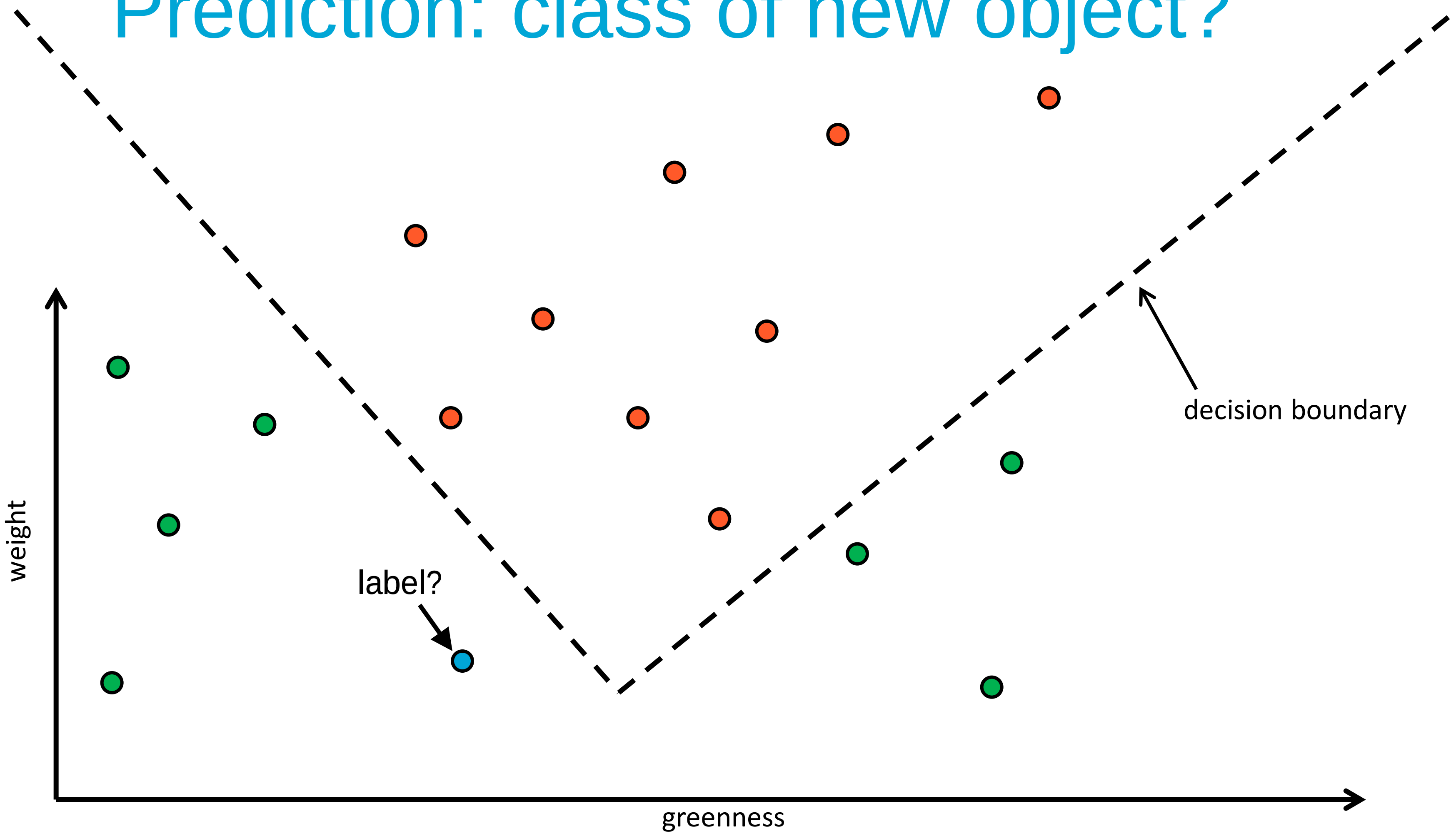
Label each object



Draw the decision boundary



Prediction: class of new object?



Two main types of machine learning

- Supervised learning
- Unsupervised learning (week 7)

Supervised learning

- The most ‘popular’ type of learning

Requirement:

Dataset with label for each training example

➔ Learns the association between example and label

Unsupervised learning

Requirement:

Unlabeled data

➔ The system learns features (= information) about the data by itself

Example of unsupervised learning tasks

- Clustering: Divides the data in clusters such that data points within a cluster are similar and those in different clusters are dissimilar

TU Delft

Example of unsupervised ML technique

- K-means clustering (week 7)

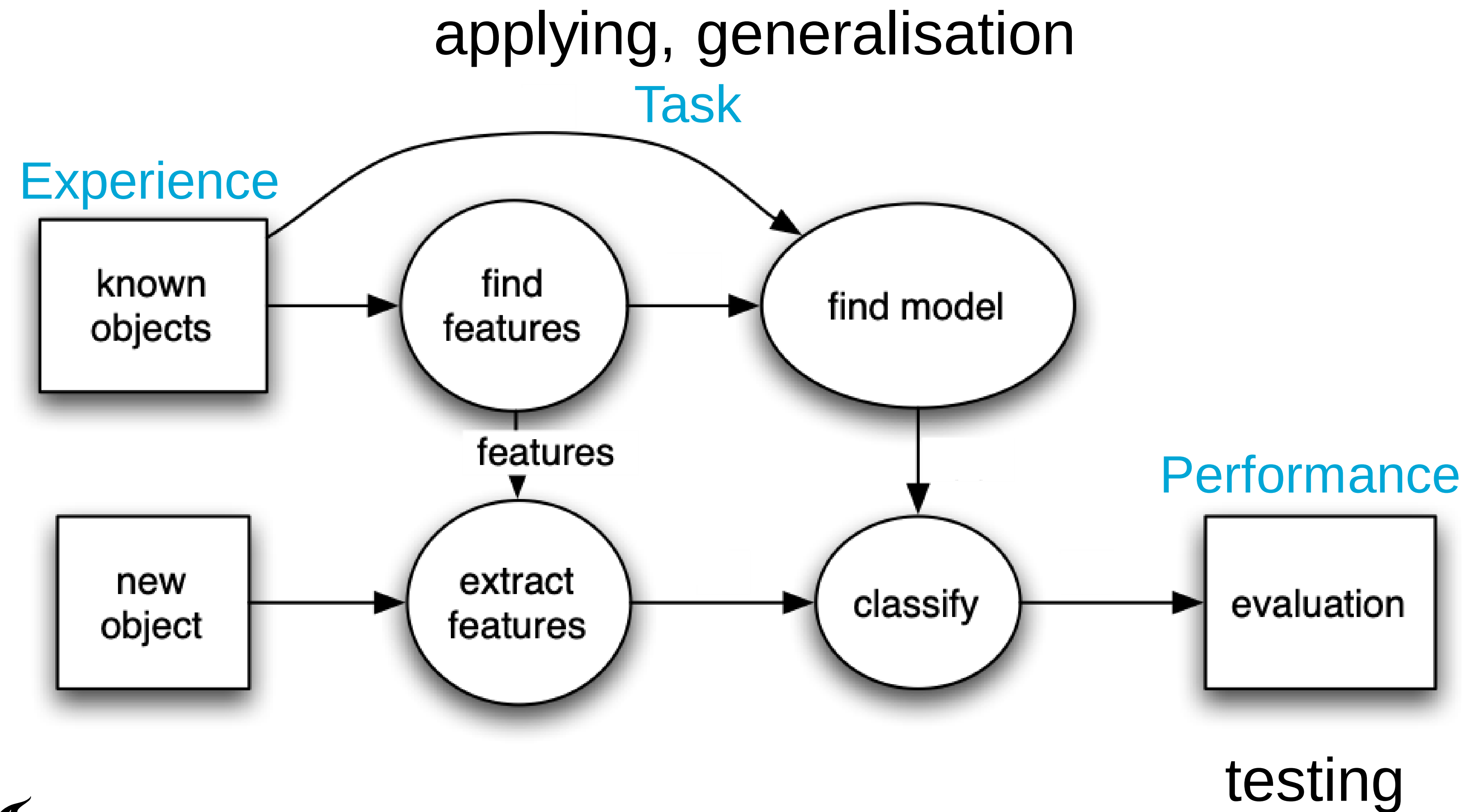
The ML pipeline

"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*."

[Tom M. Mitchell, 1997]

TU Delft

ML pipeline

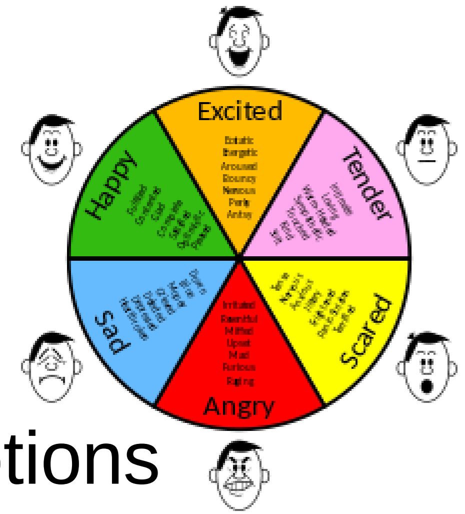

applying, generalisation
Task
Experience
known objects
find features
find model
features
Performance
new object
extract features
classify
evaluation
testing

The Task, *T*

- ML enables us to tackle tasks that are too difficult to solve with fixed programs

- Important: *learning* is the means through which we attain the ability to perform the task

➔ Learning is **not** the task



- <u>Task:</u> emotion classification
- Through *learning* how to classify emotions

Examples of supervised learning tasks

- Classification
- Regression (prediction)
- Anomaly detection
- Machine translation
- Transcription
- …

TU Delft

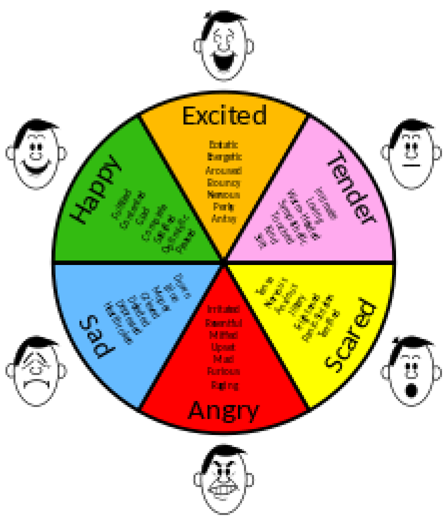
Classification: Predict a label

handwritten digits

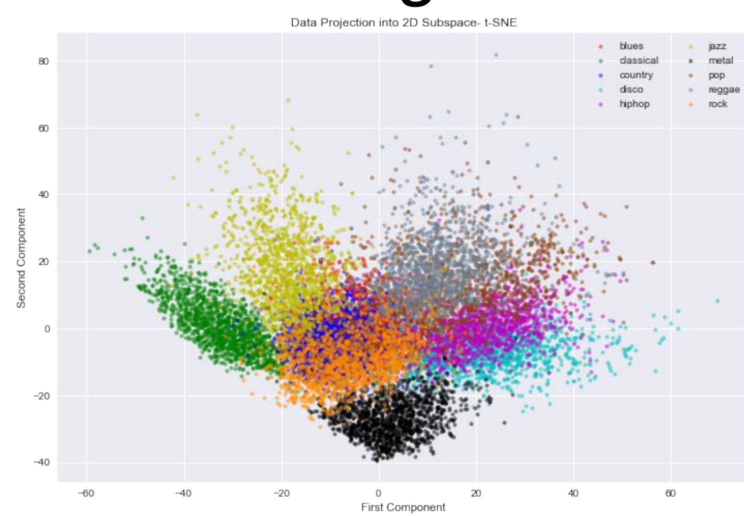
age & gender

Specify which of *k* categories some input belongs to

emotions



music genres



TU Delft

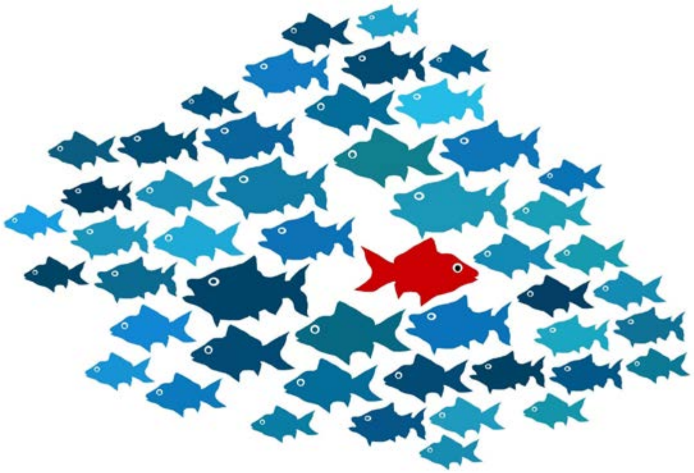
Regression: Predict a numerical value

house prices

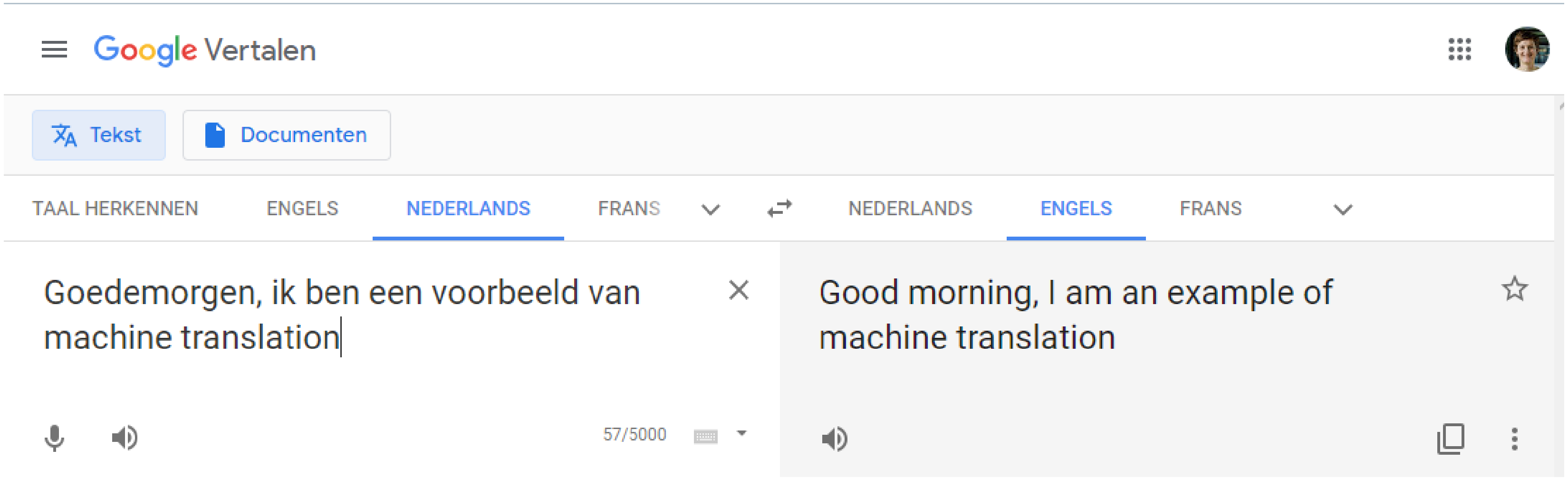
weather

Anomaly detection

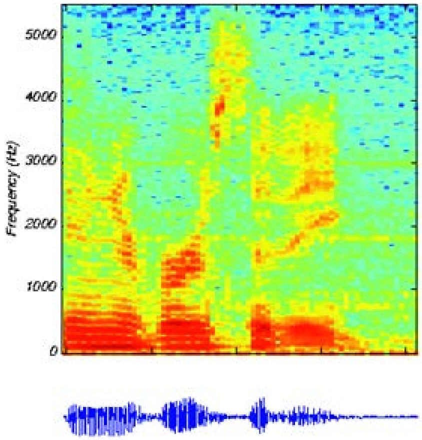
- Find unusual/atypical events or objects
- Learn and compare probability distributions



Machine translation: convert symbols in one language to symbols in another language



Transcription: Transcribe a relatively unstructured representation of some kind of data into discrete textual form



→ ASR system → “University”

automatic speech recognition

The learning

The learning to obtain the ability to carry out the task is done using one or multiple ML techniques, e.g.:

- Support vector machines (SVMs; week 4)
- Linear regression (week 4)
- Neural networks (NNs; week 6)
- Deep neural networks (DNNs; MSc course)

The performance measure, *P*

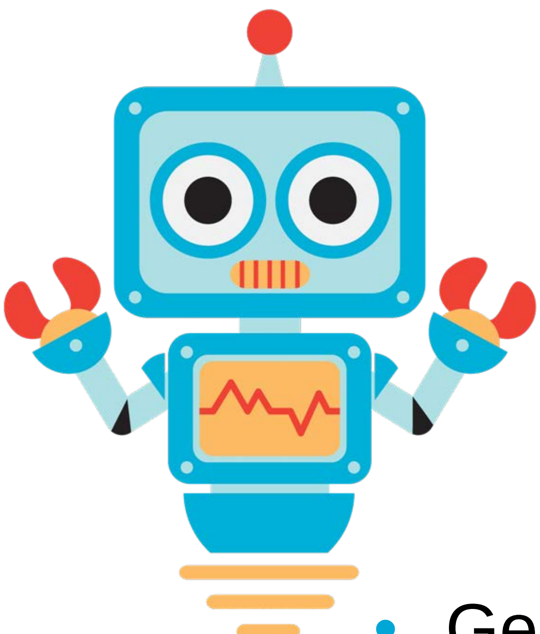
- Evaluates the abilities of the ML algorithm quantatively

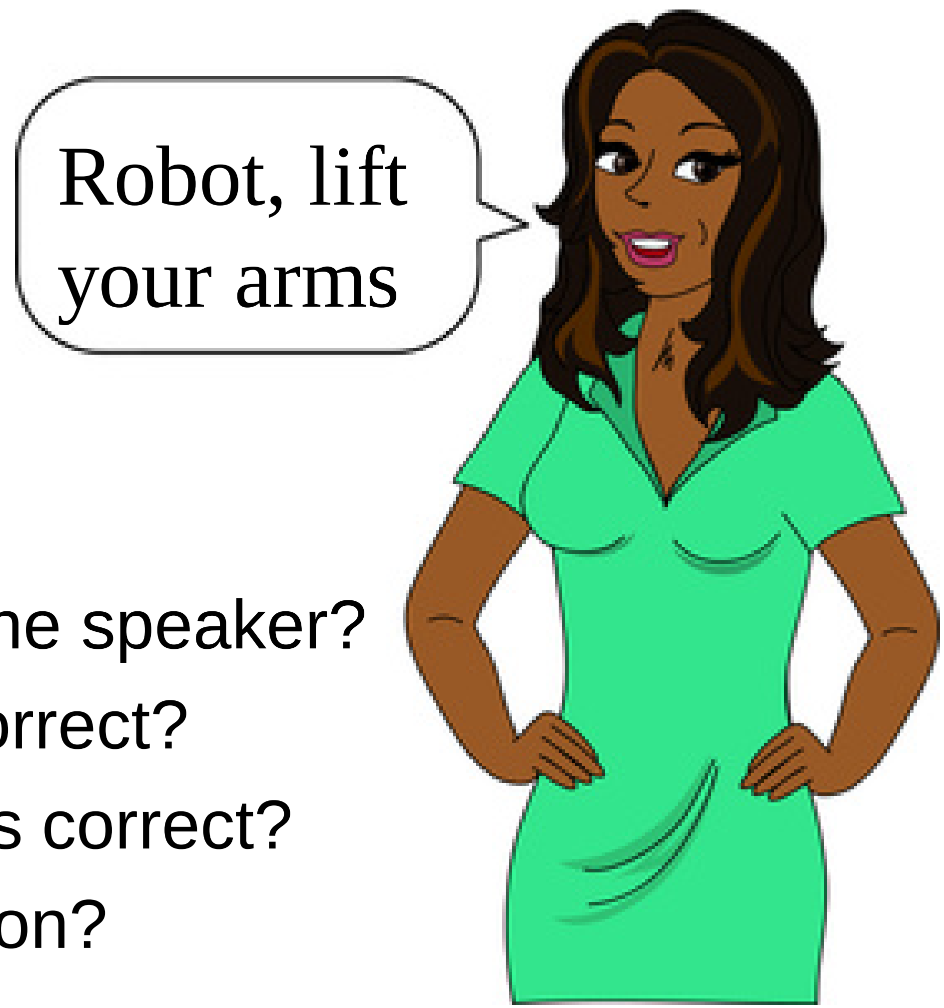
- *P* is specific to *T*
  - Classification & transcription: accuracy/error rate = proportion of correct/incorrect outputs by the model

- *P* measured on *unseen* test data
  - Data that is similar to the training data
  - But not used during training the learning algorithm

  → testing the generalisation

What performance measure to choose?





- Gender of the speaker?
- All words correct?
- Some words correct?
- Correct action?

TU Delft

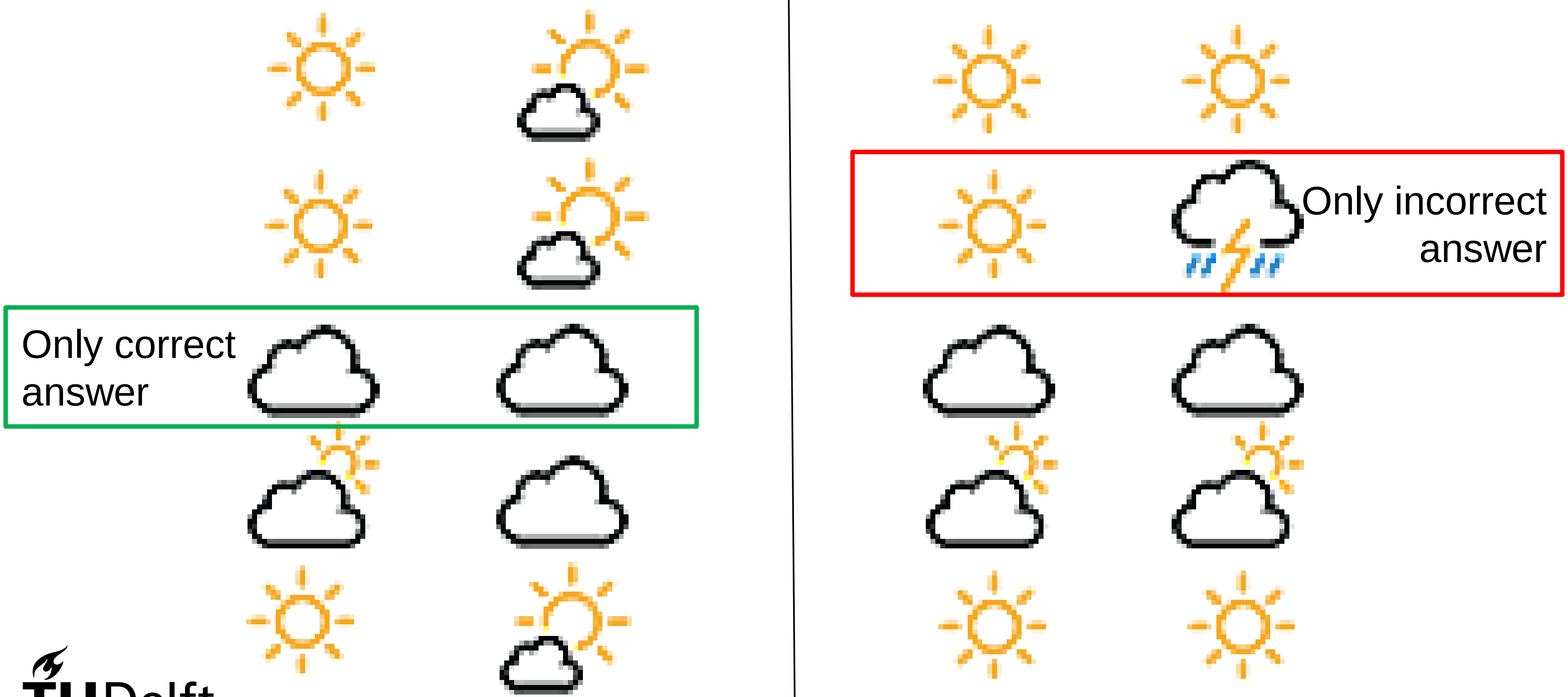
What performance measure to choose? Which is the better system?

System 1: many small mistakes

Predicted Real

System 2: one big mistake

Predicted Real



TU Delft

The Experience, *E*

== The dataset to train the ML algorithm

- Dataset: Collection of many examples or *data points*

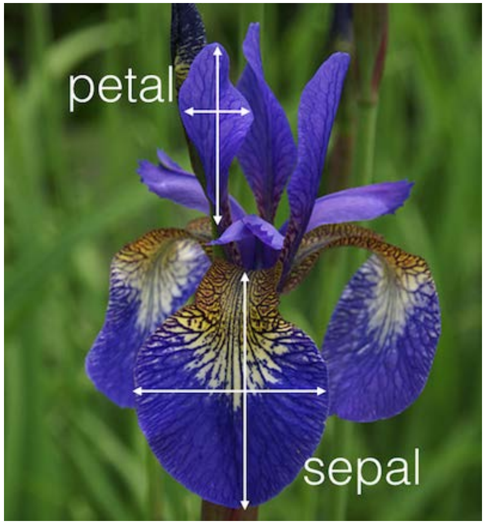
- Determines whether an ML algorithm is supervised (with labels) or unsupervised (without labels)

Iris dataset – Fisher (1936)

Iris Versicolor Iris Virginica Iris Setosa

- 150 iris plants = 150 *examples*
- 4 *features* per examples

  ➔ Measurements:
- 3 species; 1 *label* per example



Design matrix

- One example per row
- Iris dataset: 150 examples with 4 features each
- Design matrix: $X \in \mathbb{R}^{150\times4}$

where, $X_{i,1}$ is the sepal length of plant i

and $X_{i,2}$ is the sepal width of plant i, etc.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa

Experience, *E* in supervised learning

During training:

- Each example is described using features and labels (or *targets*)

- Task: classify

Iris plants into 3 species based on the measurements

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa

Experience, *E* in unsupervised learning

During training:

- Each example is described using features

- Task: e.g., clustering

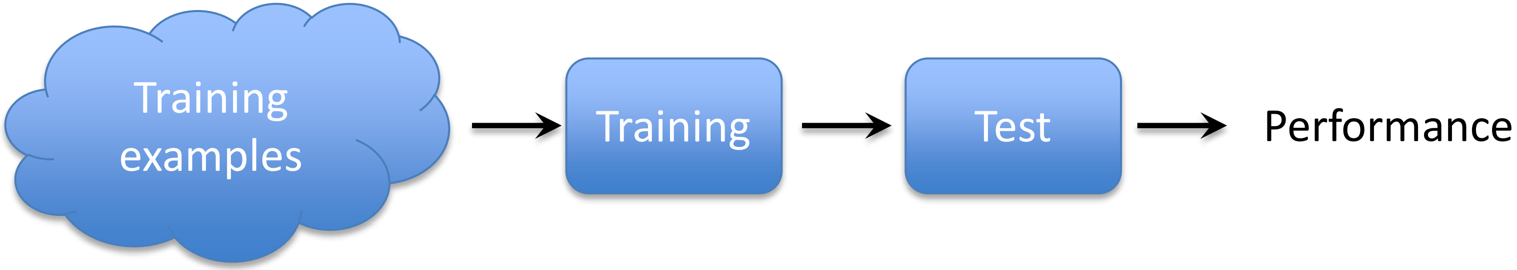
= divide the data-set into clusters of similar examples

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1
11	5.4	3.7	1.5	0.2
12	4.8	3.4	1.6	0.2
13	4.8	3.0	1.4	0.1

TU Delft

General ML pipeline

1. Train the ML algorithm using a *dataset* of *examples* with *features* for a specific *task*
2. Test the *generalisability* of the ML algorithm on an unseen testset
3. Quantify *performance* using an accurate and suitable measurement



Time to dive a bit deeper …

Supervised learning

- Learns the association between example (= input) and label (= output)

➔ By identifying patterns in the data

➔ General input-output function: $y = ax + b$

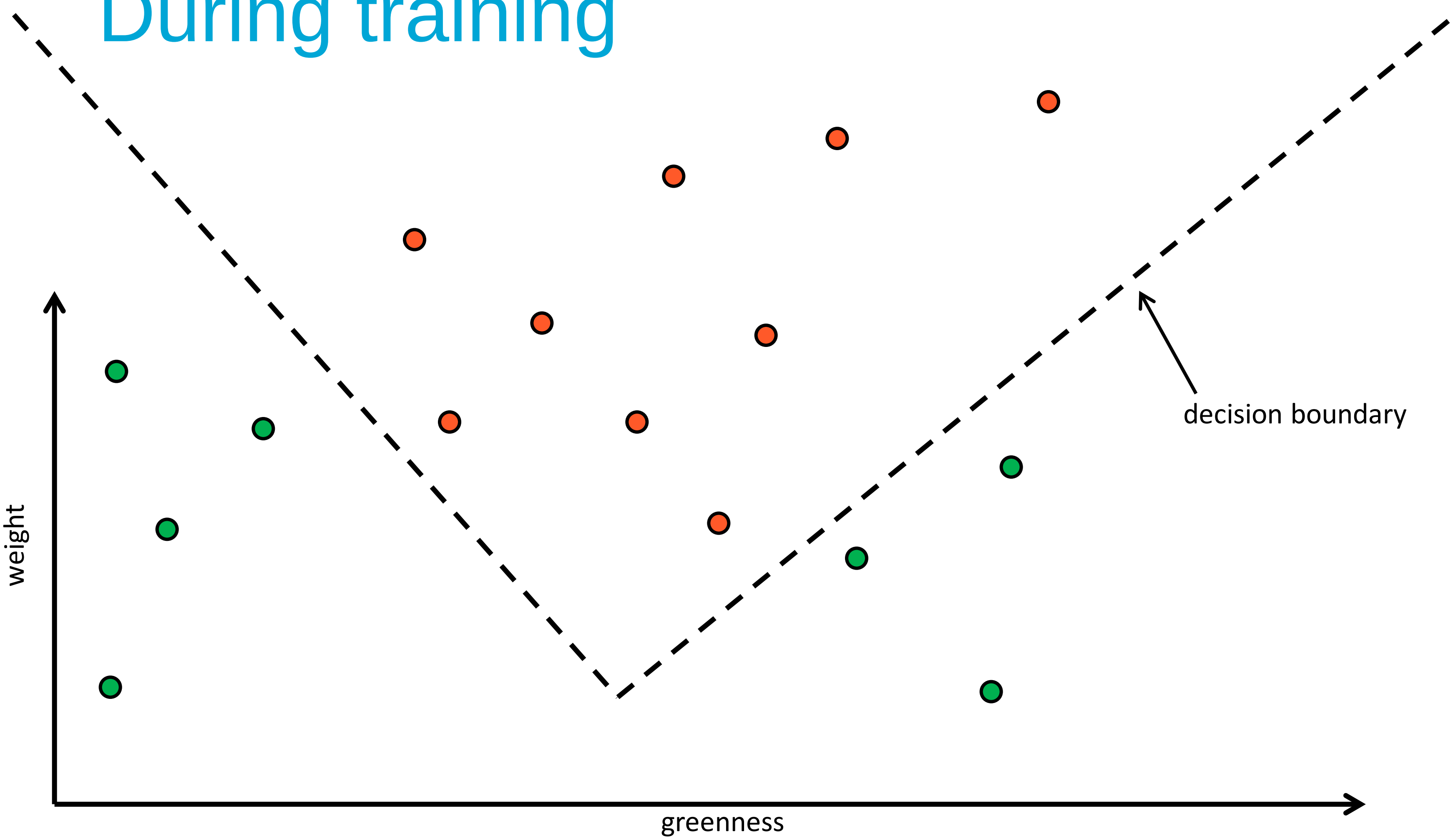
Learning = training

- General idea:
  - Collect example input-output objects (x, y objects)
  - Measure d features of choice and represent in vector space
  - Divide up feature space and assign output (or class label)

Goal of training: Learn a function that can predict a label y for a new x with as little error as possible

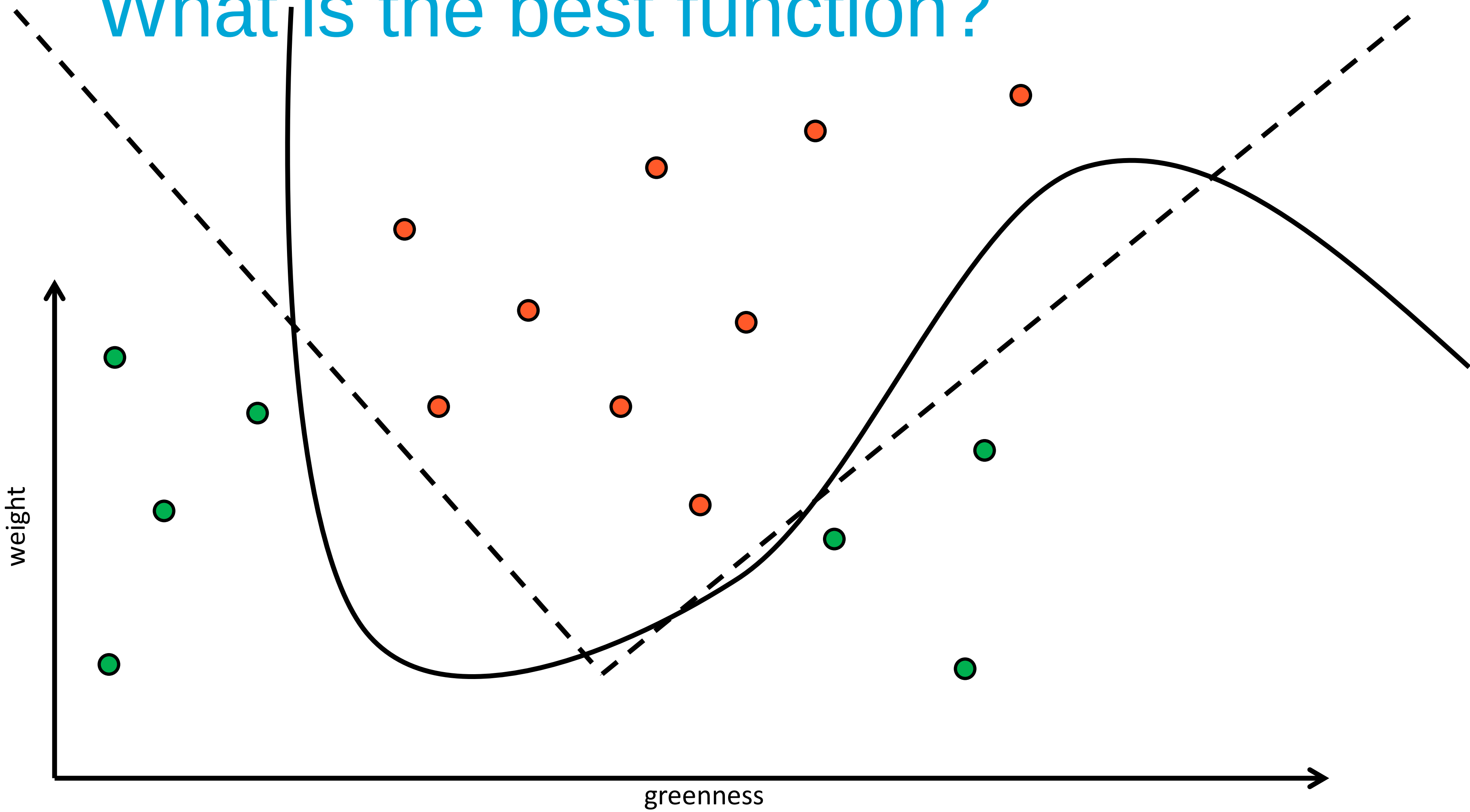
= an input-output function that can generalise to new, unseen examples (without labels)

During training



Slide adapted from Dr. Marco Loog

What is the best function?



Slide adapted from Dr. Marco Loog

Learning the optimal model parameters

- Learn model parameters a and b so that the error of the function's predictions is minimised

$$y = ax + b$$

TU Delft

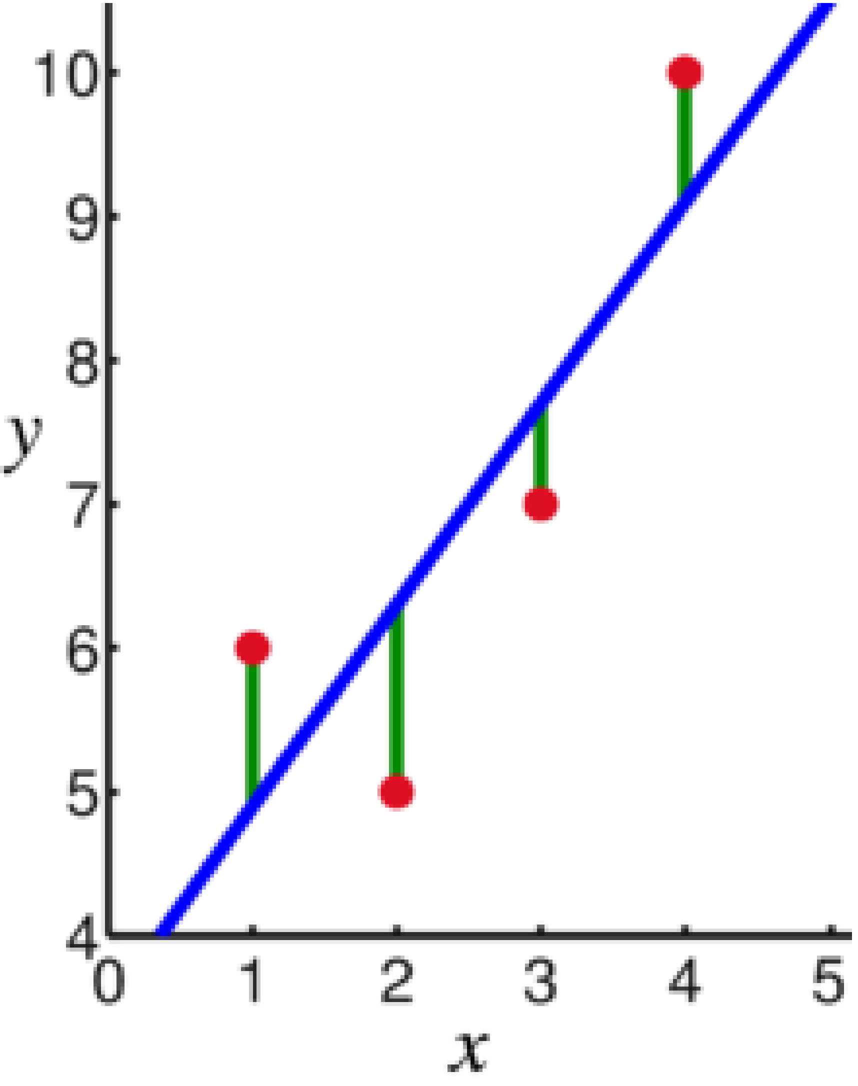
An example: linear regression

- Find a regression line

$$y = ax + b$$

that minimises the error (green lines)

Output y is a linear function of the input x



Linear regression: multiple features

- Let $\hat{y}$ be the value that the model predicts y should take:

$$\hat{y} = w^T x + b$$

- w = *set of weights* that determines how each *feature* influences the prediction $\hat{y}$
- b = intercept or bias

- Task, *T*: predict y from x by outputting
$$\hat{y} = w^T x + b$$

Performance, *P*

- Mean squared error (= Euclidean distance) of the model on the test set:

$$MSE = \frac{1}{m}\sum_i (\hat{y}^{(test)} - y^{(test)})_i^2$$

where $\hat{y}^{(test)}$ = the predictions on the test set

m = number of training examples

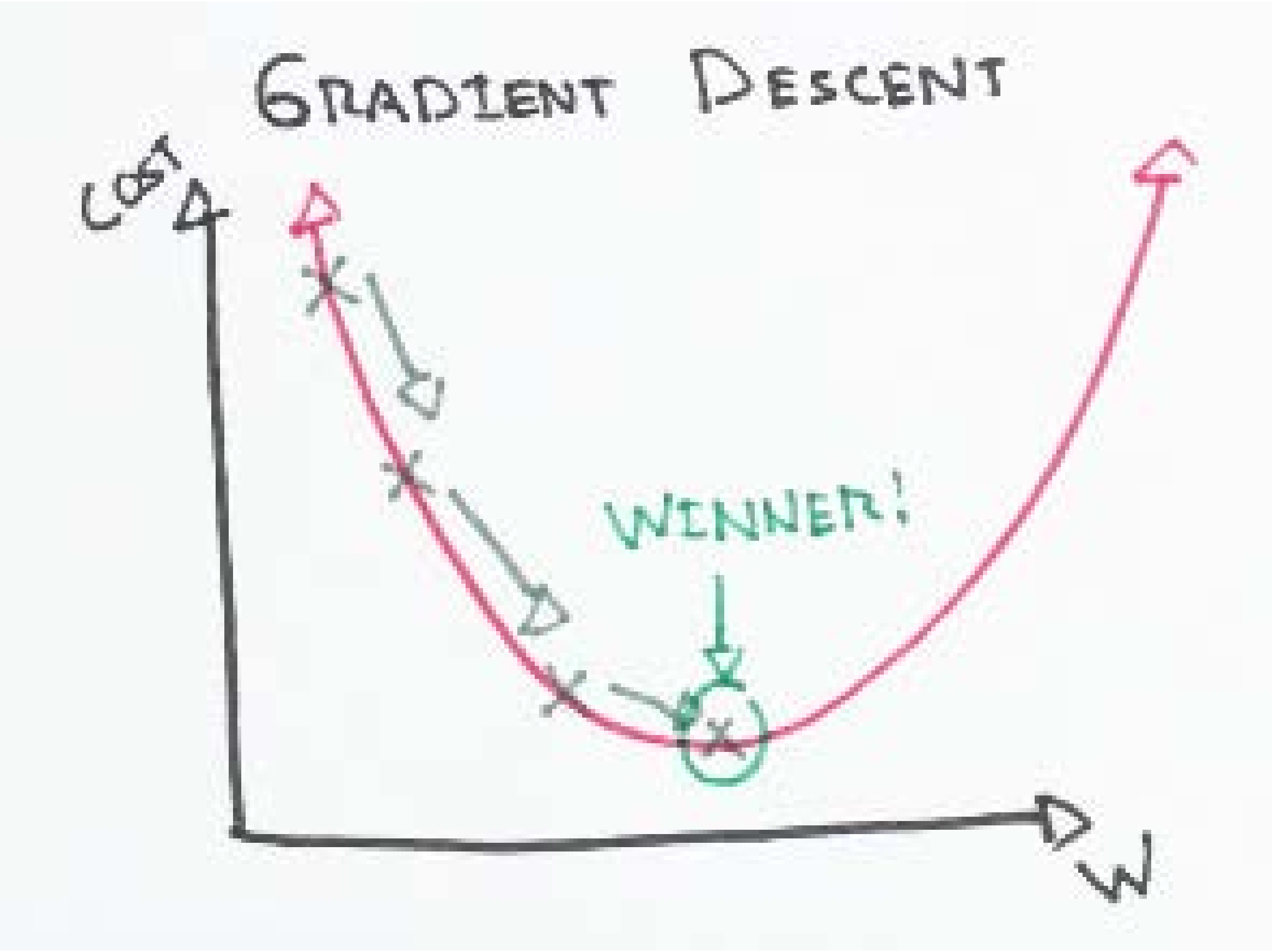
- MSE decreases to 0 when $\hat{y}^{(test)} = y^{(test)}$

Machine learning

- Set the weights w
- Minimise the error = cost = loss
- Performance, P = cost/loss function
- During training: Find the minimum of the model's loss function by iteratively getting a better and better approximation of it

TU Delft

Gradient descent



After training

- Values for w and b that minimise the error on the training data

= *optimisation* error

- We want the *generalisation* or *test* error, i.e., performance on unseen data

TU Delft

An example: linear regression

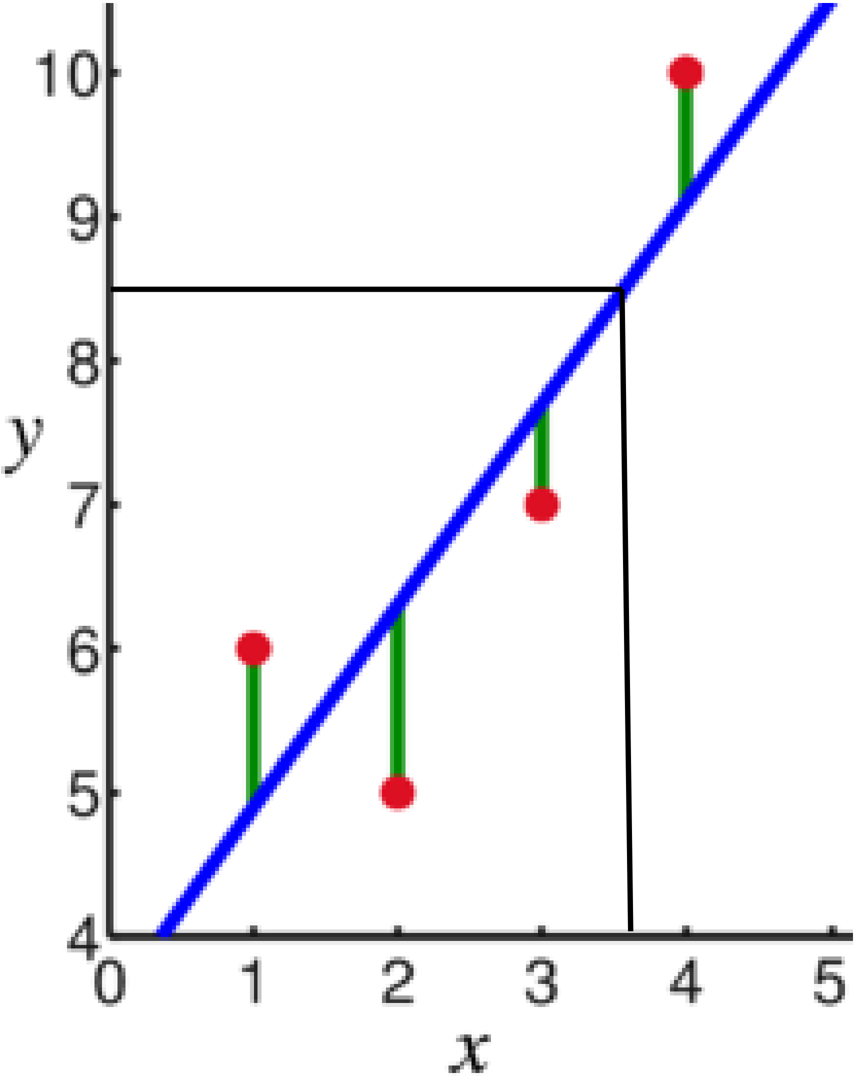
- Find a regression line

$$y = ax + b$$

that minimises the error (green lines)

Output y is a linear function of the input x

- Test: $x = 3.7$, $y = ?$



i.i.d. assumptions

- The examples in the test and training data are independent from each other
- The test and training set are identically distributed

➔ Shared underlying distribution is the data-generating distribution, p_{data}

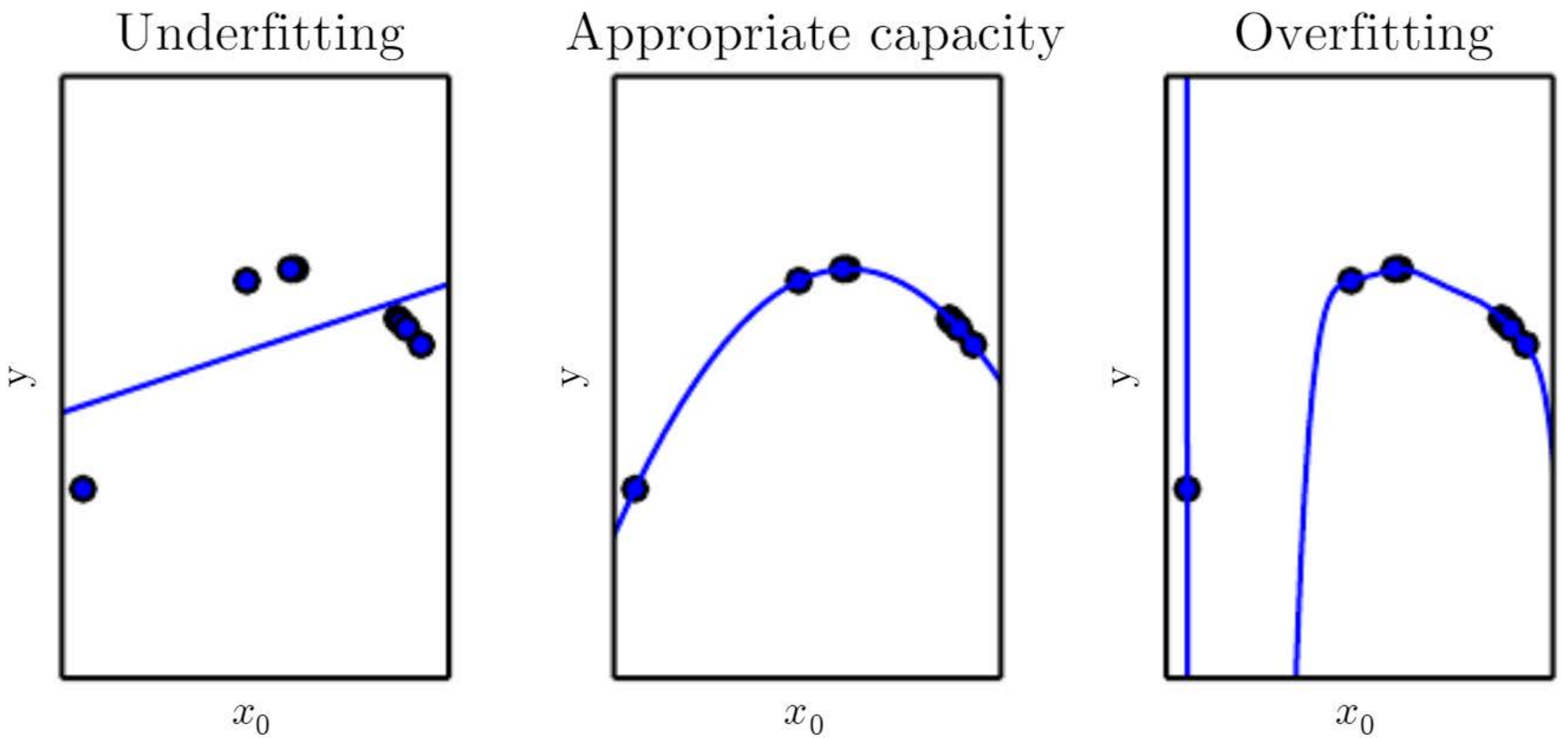
Factors that determine *P*

The ML algorithm's ability to

1. Make the training error small
2. Make the gap between training and test error small

Correspond to two central challenges in ML:

- Underfitting: training error is not small enough
- Overfitting: gap between training and test error is too big



Does not pass through all training data points

Passes through all training data points + captures the curvature in the data

Passes through all training data points but does not capture the curvature in the data

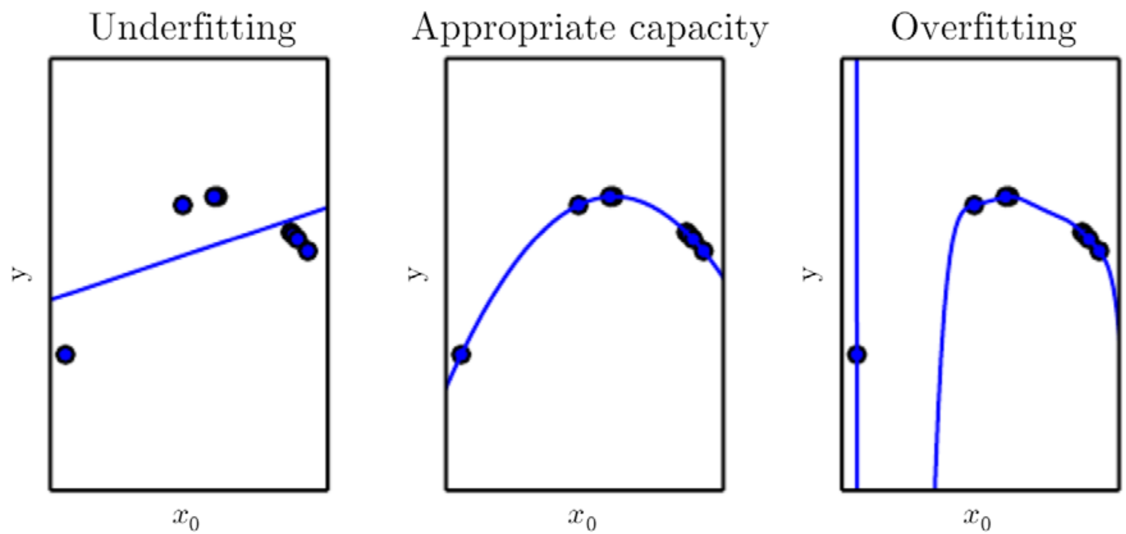
TU Delft

Figure 5.2. Deep learning book

Solution

Change the model’s capacity:

- Choose a different function type: the simplest model that has the lowest training error



Linear

Quadratic

Polynomial of degree 9

Relation between error and capacity

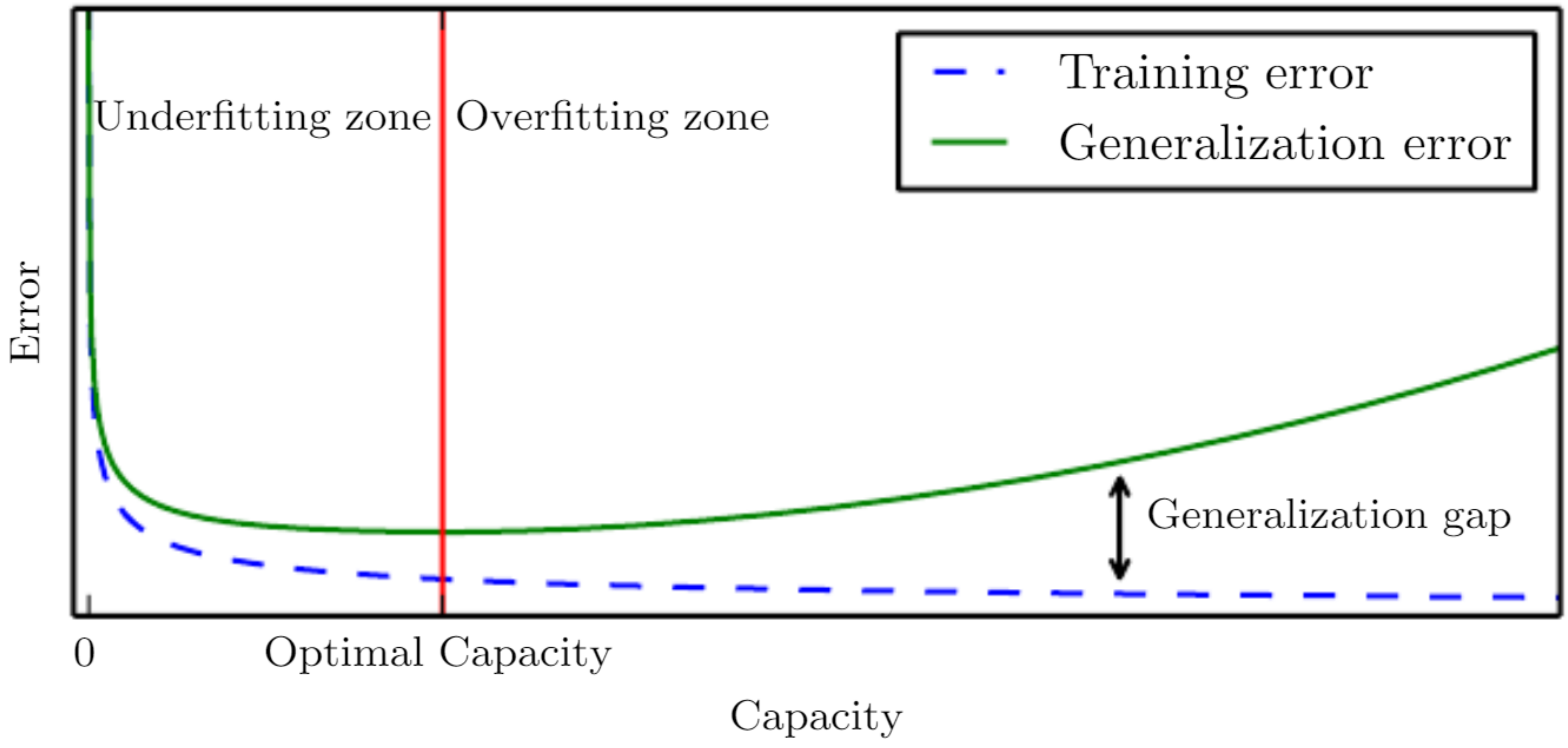


Figure 5.3. Deep learning book

Picking the right model

- Complexity control is very important (and difficult) task
  - Choose model that is not too complex but also not too simple…

- More generally: model selection is key

No free lunch theorem

- Averaged over all possible data-generating distributions, every classification algorithm has the same error rate when classifying previously unobserved points

➔ In some sense, no machine learning algorithm is universally any better than any other

No free lunch theorem

- Averaged over ***all*** possible data-generating distributions, every classification algorithm has the same error rate when classifying previously unobserved points

➔ In some sense, no machine learning algorithm is universally any better than any other

Goal of ML

- NOT to seek a universal learning algorithm or the absolute best learning algorithm

- BUT what kinds of machine learning algorithms perform well on the data drawn from the kinds of data-generating distributions we care about

→ Design ML algorithm for a specific task

Features

Earlier we said:

- Learning and predicting is based on counting the frequency of occurrence of objects
- Measure d features of choice for each object and represent in vector space

Q: What features can we choose to predict gender?

→ With more features per example, we can better tell apart the training examples

Q: What happens to the model's generalisation ability?

→ Will be discussed in more detail in the next lectures

Important notes regarding features

- Note that features give a specific view of the objects: YOU (the user) are responsible for it

- Good features allow for pattern recognition, bad features allow for nothing

➔ It is important to choose your features well!

Conclusions

- Data / Experience, E:
  - Features
  - Labels?
- Determine the Task, T
- Training = learning:
  - Choose a class of functions
  - Choose a performance measure, P
  - Optimise the function’s parameters
- Test the model’s generalisability