Kris Merckx

CONCEPTS OF DATA & ANALYTICS

Small data, big data, smart devices and AI

This book will teach you about data. How do we store and organise digital data? How can data be searched? How can it be collected?

What's more, how can this work be automated, possibly with the aid of artificial intelligence? Of course we also look at the way artificial intelligence works and is capable of making predictions.

You will learn how to visualise data in other classes ... But first I will try to set you on your way in the confusing world of (big) data and artificial intelligence.

Kris Merckx

1995. 11
1994. 11
1992 (11)
2003 (11)

Chapter 1

TEXT FILES & BINARY FILES

- **What are text files?**
- **What is the difference between text files and binary files?**
- **Why is it correct to say that most file types contain unstructured data?**

Just to be clear: the text files referred to here are not documents created in MS Word or LibreOffice even though there are some things they have in common. A file created in MS Word is a word processing file. Save it to a hard drive and it becomes a binary and possibly a compressed file.

The text files of the title refer to 'plain text' files. Files that only contain characters that can be entered on a keyboard and contain no formatting or graphical images.

As future data experts it is particularly important that you know what plain text means. As a data analyst, you will definitely be working with plain text files in text editors. What this means exactly will become clear below.

ASCII

A computer stores all data in binary format which is to say a series of zeros and ones. This is so because it is the only code most modern computers can understand. Every file consists of an often very long yet finite series of zeros and ones. The computer stores them in a storage medium (hard drive, USB stick, etc.).

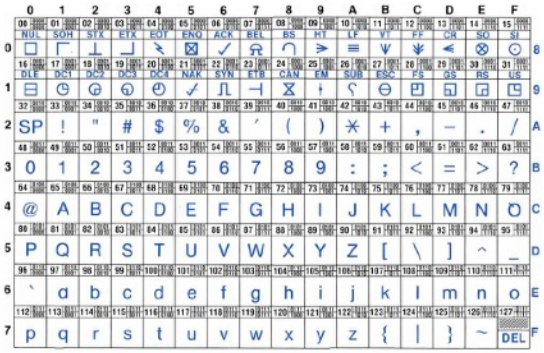
While I am typing this sentence on my keyboard, the computer is translating the keystrokes into zeros and ones, bits which in turn are grouped into bytes. One byte is a set of eight bits. The letter 'a' for instance, translates into a byte with the sequence 01100001. In decimals this is 97. Every key and combination of keys (e.g. SHIFT + A for capital A) are thus assigned a byte. Since one byte consists of 8 bits, this allows us 2^8 (2 to the power 8) of possible combinations since each bit can be either a zero or a one.

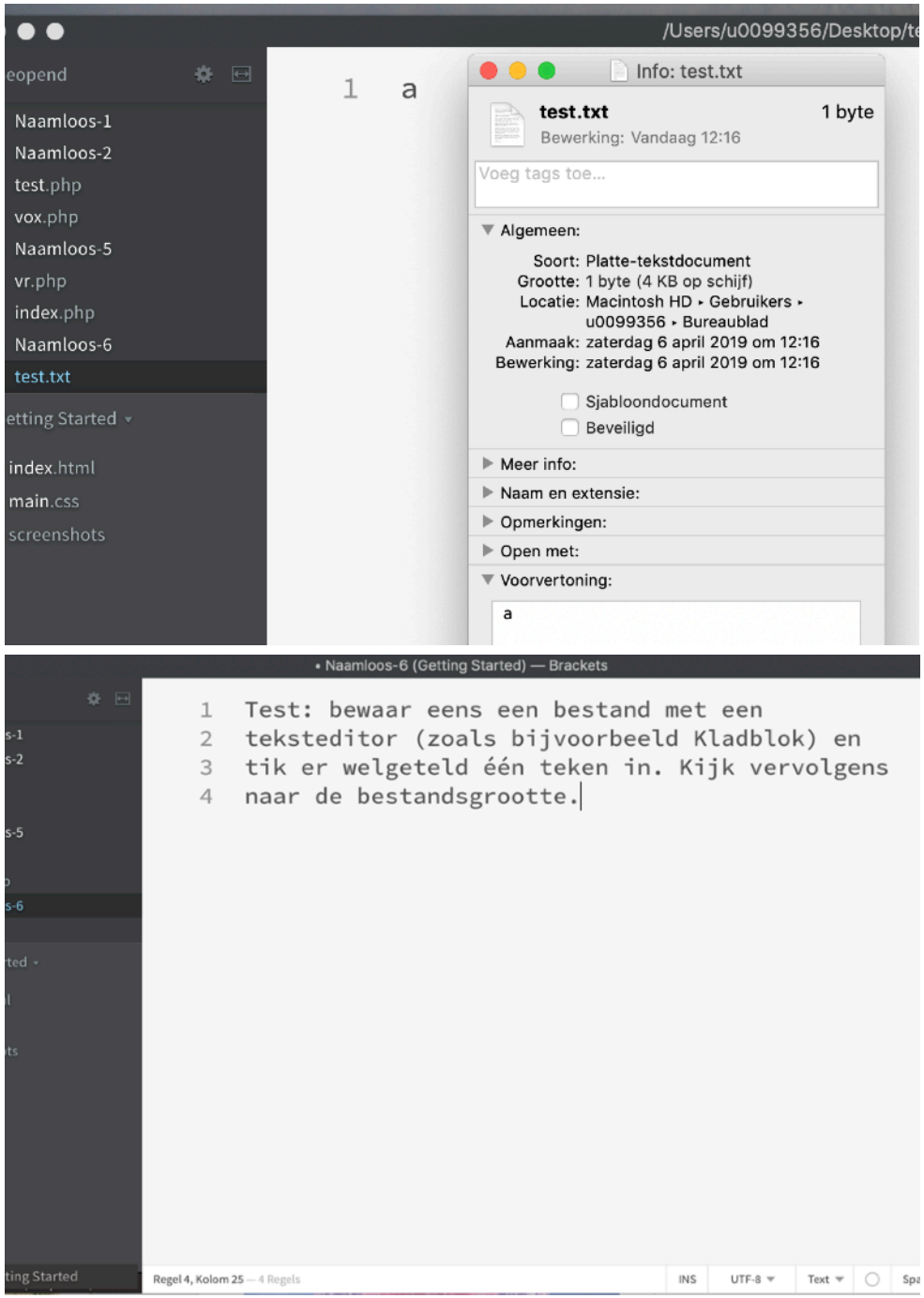
This allows for a total of 256 combinations which is to say all combinations between

```
00000000 = 0 and 11111111 = 255
```

Of course this is based on convention. In theory, you can hook up any keyboard to any digital system that allows text input. Keystrokes are recognised automatically. This is because almost all of them support ASCII. **ASCII** (American Standard Code for Information Exchange) is a standard that assigns a number to Latin letters, digits, punctuation and a number of other characters as well as a series of control characters.

The ASCII table is a two-dimensional row or tabular array. Look for the desired character and select the binary code indicated at the head of the row. Next, paste the content of the head of the column behind it. A space, for instance, hence becomes 0010 followed by a 0000, so together 00100000, which corresponds to the decimal number 32.





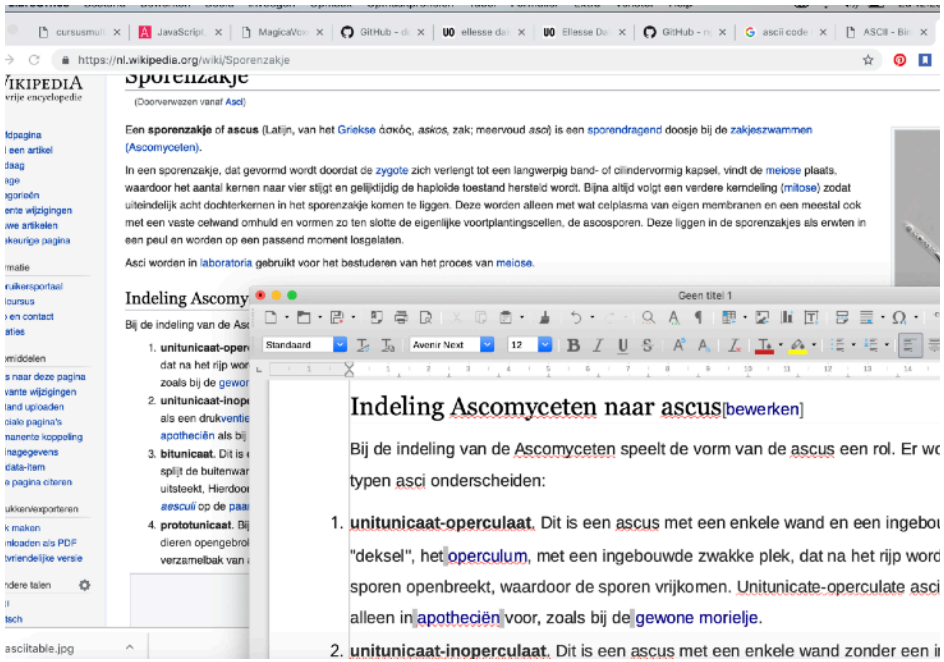
Test: save a file with a text editor (like e.g. Notepad) and type precisely one character. Then have a look at the size of the file (see illustration to the left)

A text editor only allows you to enter ASCII characters (or UTF-8 characters, more about these later). It doesn't allow you to format text or add pictures, for instance. I can already hear you think: so what's the point of that?

The greatest advantage is that these types of files are perfectly exchangeable. There is no special software required to read the content of these documents. A simple editor like Windows Notepad or Adobe Brackets suffices to open the file. Programming code or webpages can easily be opened in an editor because they only include characters supported by the ASCII or UTF standards.

Illustration: text in the editor Adobe Brackets

You may have experienced this yourself. You copy a text from Wikipedia and paste it into a Word document. All links included in the webpage as well as the formatting were copied along. This can be avoided by pasting the text into a text editor first. That way, the only thing left is plain text.



All links included in the webpage as well as the formatting were copied along.

UNICODE and UTF-8

As you can see, ASCII works like charm... except for anyone availing themself of a language other than a standard (Indo-)European oriented language like Arabic, Chinese or Japanese. There were no binary codes for Arab or Chinese (a Chinese average Joe knows about 7,000 different characters). Even modern Hanyu Cidian Chinese comprises some 56,000 characters! Extensions to the ASCII character set were required. The necessity to include all other language systems and codes led to UNICODE.

The **UNICODE** character set allows the use of all possible alphabets and writing systems in a single document. Even if it is unlikely that you will find yourself writing in Pau Cin Hau, Bamum or Sharada on a regular basis, you are more likely to cross paths with dingbats or mathematical symbols. UNICODE is an ISO standard.

The best known UNICODE character set is **UTF-8** (8 bit Unicode Transformation Format). Modern webpages are coded in this standard and can consequently display any symbol by default.

Binary files

Even though there are many advantages to ASCII files, it is not always the best way to store data.

When we talk about BINARY files, we are referring to files that were not encoded using the ASCII character set. *There are several good reasons why to go this way.*

Most files like pictures, documents from word processors, presentations, films, etc. are stored using 'binary coding'. Such files coded in ASCII would be too large.

Very often compression techniques are used to make files considerably smaller so that they occupy less space. Just consider how quickly your smartphone fills up when you shoot pictures or video all day.

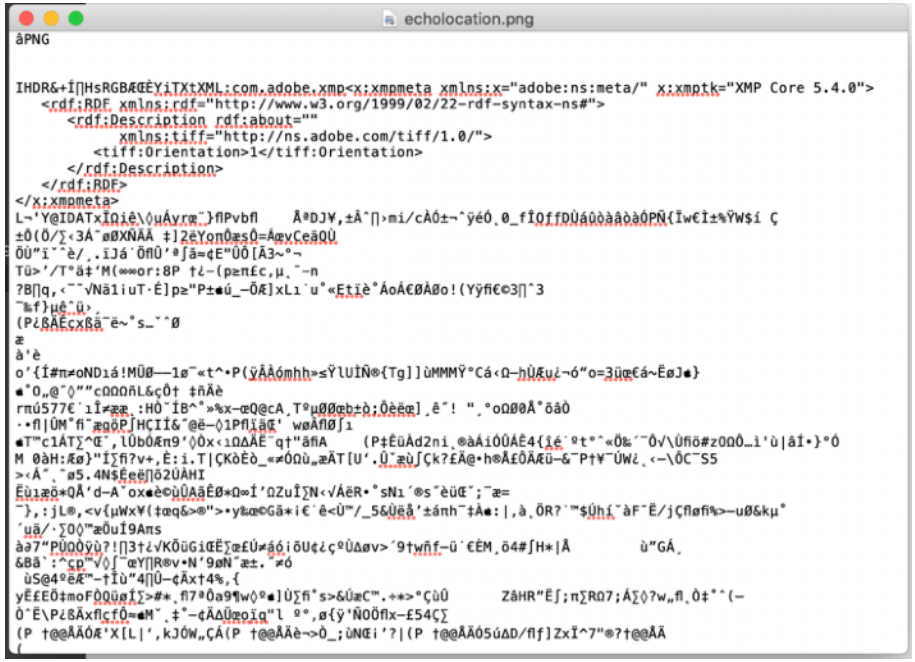
Why not ASCII?

An example: the letter ‘a’ (not capitalised) on your keyboard is the equivalent to the byte 01100001. A single character is replaced by a series of 8 zeros and/or ones. So in this particular case, a computer uses eight bits where a human being only requires one. The same is true for the digits on your keyboard. When you type a ‘0’, according to ASCII rules this becomes 00110000 and a ‘1’ becomes 00110001. The number 256, ASCII considers as three separate ASCII characters or three bytes which will be rendered as follows: 00110010 00110101 00110110.

If the only thing we want to do is store a number, this is a kind of a funky way of doing things since internally, ASCII is rendering to decimals by linking a decimal number to every key or character.

Numbers can be stored in a much more simple and compact way by not storing them as a series of separate characters but in their entirety as a number. 255 will then look like 11111111. A single byte can hold 256 different values. With 4 bytes there are 4 billion possible combinations. So the number 4,000,000,000 (4 billion) would only take up 4 bytes in this scenario. If we were to store it in ASCII, it would chew up 10 bytes.

This is the reason many different forms of coding are used. There would be no sense in storing a picture, for instance, as a series of ASCII characters (albeit digitally coded). A picture of 1,024 by 768 pixels has a total of 786,432 pixels. Since each pixel consists of a mix of 256 red values, 256 green values and 256 blue values, coding with the ASCII standard would take an enormous amount of space.



A representation of a binary file (PNG image) in a text editor

Knowing, understanding

1. Describe clearly and concisely what ASCII files are.
2. Explain the necessity of UNICODE.
3. Explain the concepts of ASCII, UNICODE and UTF-8.
4. Explain why text coding is not always the most suitable storing technique.
5. When are files called binary files?

Doing

1. Being able to categorise files as binary files or text files.

YOUTUBE
POPULAR
FRIENDS
WORDPRESS
COMMUNICATION
GOOGLE+
FACEBOOK
MEDIA
WEB
MOBILE
FLICKR
SEARCH
GLOBAL
COMMUNITY
INFORMATION
PEOPLE
LINKEDIN
PINTEREST
SHARE
CONTENT
INTERNET
MONEY
LIKE
ONLINE
NETWORK
SOCIAL
RADIO
DATA
NEWS

Chapter 2

STRUCTURED AND UNSTRUCTURED DATA

- **What is the difference between structured and unstructured data?**
- **How can data be structured?**
- **When is data semantic?**

Most data and files contain what computers interpret as rather unstructured data. That sounds strange when you realise that digital data is created for and by computers.

However, this doesn't mean that a computer also understands what the data represents. A picture, for instance, is put together by a computer as a grid of columns and rows to be filled with colour information. To a computer, a pixel (picture element) constitutes one cell in such a grid and contains three number values: a red value, a green value and a blue value. In addition, a computer will attribute an x and a y value to each pixel which makes for a total of five number values for each pixel.

So that is structured in a sense, but it says nothing about what the picture shows. Does it show a cow, an apple or a bottle of beer? The data doesn't say anything about that in this case. This information could be added so that the image source is no longer limited to pixel values and positions but also includes tags.

The same holds true for texts. Every letter in a text is stored as a number code, but the computer has no information as to the content of the text.

Structured data

So what do we mean by structured data? Structured data is:

- data with clear patterns that a machine/computer can easily recognise. For instance: a list with rows and columns, credit card numbers, postal codes, ...
- data with key and value pairs.
- For instance: structured messages on bank transfers or a JSON file

Unstructured data

Unstructured data offers little or no clue as to the content of the data or the “semantics”.

- Data that is not listed in a table with a recognisable number of rows and columns or keys with their value.
- Data with patterns that are less clear to computer systems (which is not to say there is no pattern at all).
- For instance: text documents, webpages, presentations, etc.
- For instance: pictures, audio, video, etc.

Types of structured data

Over time, a number of standards have been developed for structuring text data. Markup languages like HTML (hypertext markup language) and XML (extensible markup language) mark elements in a text file by indicating where a part starts and where it ends. It allows you to mark various parts of your data which in turn allows software to retrieve the location of a certain element faster.

Text data can also be separated by spaces, tabs or punctuation. A CSV file separates rows with line breaks (hard returns) and columns with a comma. YAML (YAML Ain’t Markup Language) separates data with tabs and line breaks. JSON (Javascript Object Notation) uses various punctuation marks (commas, quotation marks, etc.) and curly brackets.

An XML file with information on books.

```xml
<?xml version="1.0"?>
<catalog>
   <book id="bk101">
      <author>Gambardella, Matthew</author>
      <title>XML Developer's Guide</title>
      <genre>Computer</genre>
      <price>44.95</price>
      <publish_date>2000-10-01</publish_date>
      <description>An in-depth look at creating applications
      with XML.</description>
   </book>
   <book id="bk102">
      <author>Ralls, Kim</author>
      <title>Midnight Rain</title>
      <genre>Fantasy</genre>
      <price>5.95</price>
      <publish_date>2000-12-16</publish_date>
      <description>A former architect battles corporate zombies,
      an evil sorceress, and her own childhood to become queen
      of the world.</description>
   </book>
</catalog>
```

A YAML file with data on members of staff.

```yaml
# Employee records
-  martin:
    name: Martin D'vloper
    job: Developer
    skills:
      - python
      - perl
      - pascal
-  tabitha:
    name: Tabitha Bitumen
    job: Developer
    skills:
      - lisp
      - fortran
      - erlang
```

A CSV file with address data. Most spreadsheet programmes like MS Excel can import and export CSV data. It's a very common format for data exchange.

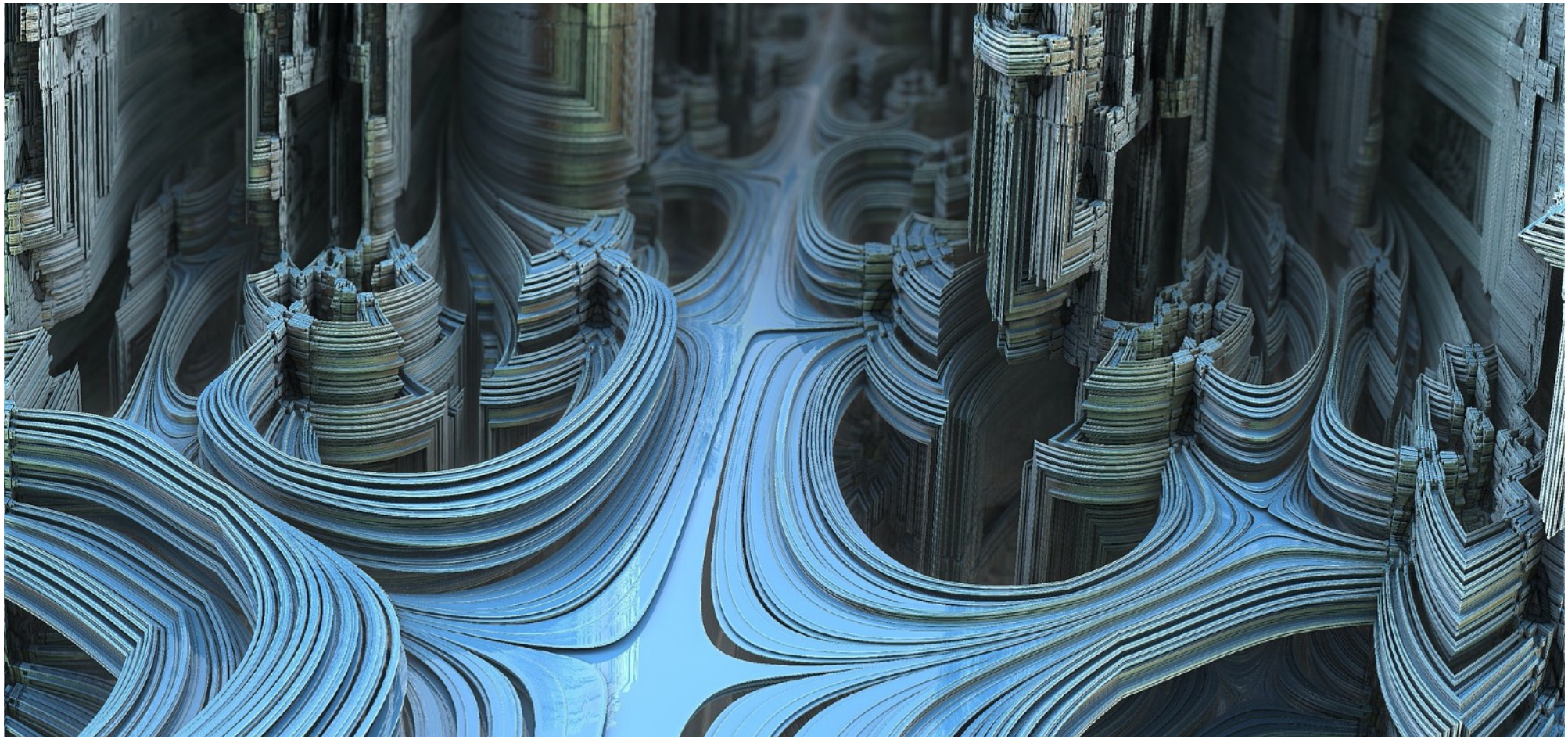
```
Naam,Functie ,Telnr.,Mob,E-mail,Web,Aantallen,,
Jeroen Trogh,Head Of Operations Academy,+32 (0) 16 22 85 08,+32 (0) 485 84
80 32,jeroen.trogh@ohl.be,www.ohleuven.com,40,,
Ben Geukens,Head Of Operations Academy,+32 (0) 16 22 85 08,+32 (0) 478 18
87 54,ben.geukens@ohl.be,www.ohleuven.com,40,,
Henk Mariman,Academy Manager,+32 (0) 16 22 85 08,+32 (0) 475 58 41
26,henk.mariman@ohl.be,www.ohleuven.com,40,,
Gauthier Vanderloock,Video & Data Analyst U18-U21,+32 (0) 16 22 85 08,+32
(0) 490 39 62 90,gauthier.vanderloock@ohl.be,www.ohleuven.com,40,,
```

An ICAL file for exchanging calendar data.

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//ZContent.net//Zap Calendar 1.0//EN
CALSCALE:GREGORIAN
METHOD:PUBLISH
BEGIN:VEVENT
SUMMARY:Abraham Lincoln
UID:c7614cff-3549-4a00-9152-d25cc1fe077d
SEQUENCE:0
STATUS:CONFIRMED
TRANSP:TRANSPARENT
RRULE:FREQ=YEARLY;INTERVAL=1;BYMONTH=2;BYMONTHDAY=12
DTSTART:20080212
DTEND:20080213
DTSTAMP:20150421T141403
CATEGORIES:U.S. Presidents,Civil War People
LOCATION:Hodgenville\, Kentucky
GEO:37.5739497;-85.7399606
DESCRIPTION:Born February 12\, 1809\nSixteenth President (1861-1865)\n\n\n
 \nhttp://AmericanHistoryCalendar.com
URL:http://americanhistorycalendar.com/peoplecalendar/1,328-abraham-lincol
 n
END:VEVENT
END:VCALENDAR
```

A JSON file according to a shared schema (www.schema.org)

```
{
    "@context": "http://schema.org/",
    "@type": "TravelAgency",
    "address": {
        "@type": "PostalAddress",
        "addressCountry": "Belgium",
        "addressLocality": "Leuven",
        "postalCode": "3000",
        "streetAddress": "Quinten Metsysplein 12",
        "email": "cultuurreizen@davidsfonds.be",
        "telephone": "003216310608"
    },
    "email": "cultuurreizen@davidsfonds.be"
}
```



Knowing, understanding

1. Being able to explain the key characteristics of structured data.
2. Being able to explain the key characteristics of unstructured data.
3. Being able to list the main types of structured data.

Doing

1. Recognising files types: JSON, XML, HTML, YAML, CSV, ICAL...
2. Being able to distinguish structured and unstructured files.
3. Being able to identify a file according to the above categories.

Chapter 3

DATABASES

- What are databases?
- What is the difference between databases and other computer files?
- How can you store information in databases?
- What kind of databases are there?

A database allows data to be stored in a structured way. This makes it possible to retrieve data quickly.

How do databases work? What kind of databases are there? How can you use software to query databases and search data?

Relational tables

If you have always believed this to be true, more is the pity, but a spreadsheet programme like MS Excel is not a database application, even though many people use it as such. Databases hold structured information by putting it in one or more tables.

Do you do your shopping at stores that issue customer cards? The first thing they often ask when buying something is your postal code. Below you will find out why that is.

Most people would keep a simple address list in a single table that would look something like this:

Last name	First name	Postal code	Municipality	Street
Jansens	Jan	3300	Tienen	Bostsestraat 2
Pieters	Peter	3000	Leuven	Naamsestraat 13
Anders	Ann	3001	Heverlee	Leo Dartelaan 10
...	...	...	...	...

A simple database like this often doesn't suffice. Also in the case of the above example there is room for improvement: as your address list grows, it's likely you will often have to enter the same municipality and postal code and the same street name. We could split it into 3 tables.

1 A table for the municipalities

ID	Municipality	Postalcode
1	Tienen	3300
2	Leuven	3000
3	Heverlee	3001
...	...	...

Note that one column is tagged 'ID'. An ID is a unique number. You can arrange it so that the database automatically raises the number by one when adding a row. If a row is removed the other numbers remain the same. This may seem odd, but it is quite normal. Think of it as a tracking number or a shirt number. Cyclists don't change shirt numbers when one of them falls out of the race.

2 A table for the streets

We will furthermore set up a separate table for the street names. Every municipality has several streets, but any one street is (generally) located in one municipality (of course that is not always true).

In this case the table contains 3 columns: an ID, the ID of the municipality from the first table and the street name.

ID	Streetname	IDMunicipality
1	Bostsestraat	1
2	Naamsestraat	2
3	Leo Dartelaan	3
4	Putstraat	1
5	Houbaertstraat	1
6	Bondgenotenlaan	2
…	…	…

The table of municipalities and the table of streets are intrinsically related. You can now 'filter' and search data quickly. For instance: list all streets of Tienen. In database terms, this comes down to the following: select all fields in the table "Streets" for which IDMunicipality = 1.

A little more advanced, you can use the following query: *list all streets of the municipality with postal code 3300*, which is the postal code of Tienen.

3 A table for the contacts

LastName	FirstName	StreetNumber	IDStreet
Jansens	Jan	2	1
Pieters	Peter	13	2
Anders	Ann	10	3
…	…	…	…

By connecting the three tables, you can find out who lives in which municipality and on what street.

For instance: Jan Janssen lives on IDStreet 1. This ID can be found in the table with the street names. This table contains the ID of the municipality in the table called "municipalities".

It may seem a roundabout way of doing things, but in the end it makes your data much more **structured** and more **compact**. It furthermore ensures **data integrity** and avoids different spellings and errors.

Editing databases with CRUD

Of course databases must also be easy to edit. It has to be possible to add elements to tables, remove data, modify information. In most cases, you simply want to search, filter and read the information of a database.

CRUD stands for Create, Read, Update, Delete.

C	Create	Persistent storage of data. Data cannot have doubles. IDs, for instance, must persist. If you remove a row, all the other rows keep their original ID. Otherwise the "relations" are botched.
R	Read	Simple searching and filtering of data.
U	Update	Modifying data in the database.
D	Delete	Removing data from the database.

Of course databases require **user rights**. Not everyone should be allowed to erase or edit data, or even add data. Applications that do allow this, like Wikipedia, keep an edit history and have a built-in control mechanism to avoid abuse or errors.

Databases speak SQL

In order to carry out CRUD commands, most databases support the language **SQL (Structured Query Language)**.

If we were to filter the data from our address list on street in SQL, this is what that would look like:

*SELECT*FROM contacts WHERE IDStreet=1*

*DELETE*FROM streets WHERE IDMunicipality=2*

Data types

Last but not least, the columns or "fields" (a cell in a table) in a database table contain certain types of data: some may only contain integers ("int" or "integer"), others text (a "string" with predetermined length), a date, etc. These data types are also important for ensuring data integrity.

Relational databases

The example above is based on the assumption that any given person only has one house number. The street name doesn't need to be filled in. We need only refer to a unique ID (identity) given to the street in the table "streets". Each street can only be in one municipality. But there can be several municipalities that have a "High Street", I hear you say. This is irrelevant here. From the moment you set Street_ID=1, the database knows that Street_ID=1 is located in Municipality_ID=1. This way, you are automatically directed to the correct municipality.

In the user interface, the first enquiry may be for the postal code. Once that is selected or entered, you then select the municipality. Next is a list of all the streets in that municipality. When you select one, you will be presented an overview of all your addresses in that street. An additional advantage is that when a street changes names, the street name is automatically updated for all addresses.

Because these databases have techniques to link different tables together, to create a relationship between them (e.g. ID_municipality, ID_Street...) they are called RELATIONAL DATABASES.

Relational databases are used extensively in companies, webshops, service industries, websites, electronic learning environments and so on. Such databases allow for content offered by users to be stored in a meaningful way.

- Student lists.
- Score lists in a digital learning environment.
- Products, brands, colours and sizes in a webshop.
- Invoices, payments and tenders in an accounting programme.
- Social network sites like Facebook and Twitter and search engines like Google, store their results and index in large databases.
- Stock keeping.
- etc.

In a great many cases, software is still using traditional **relational database management systems** or **RDBMS**. However, they are not always the best solution in all situations.

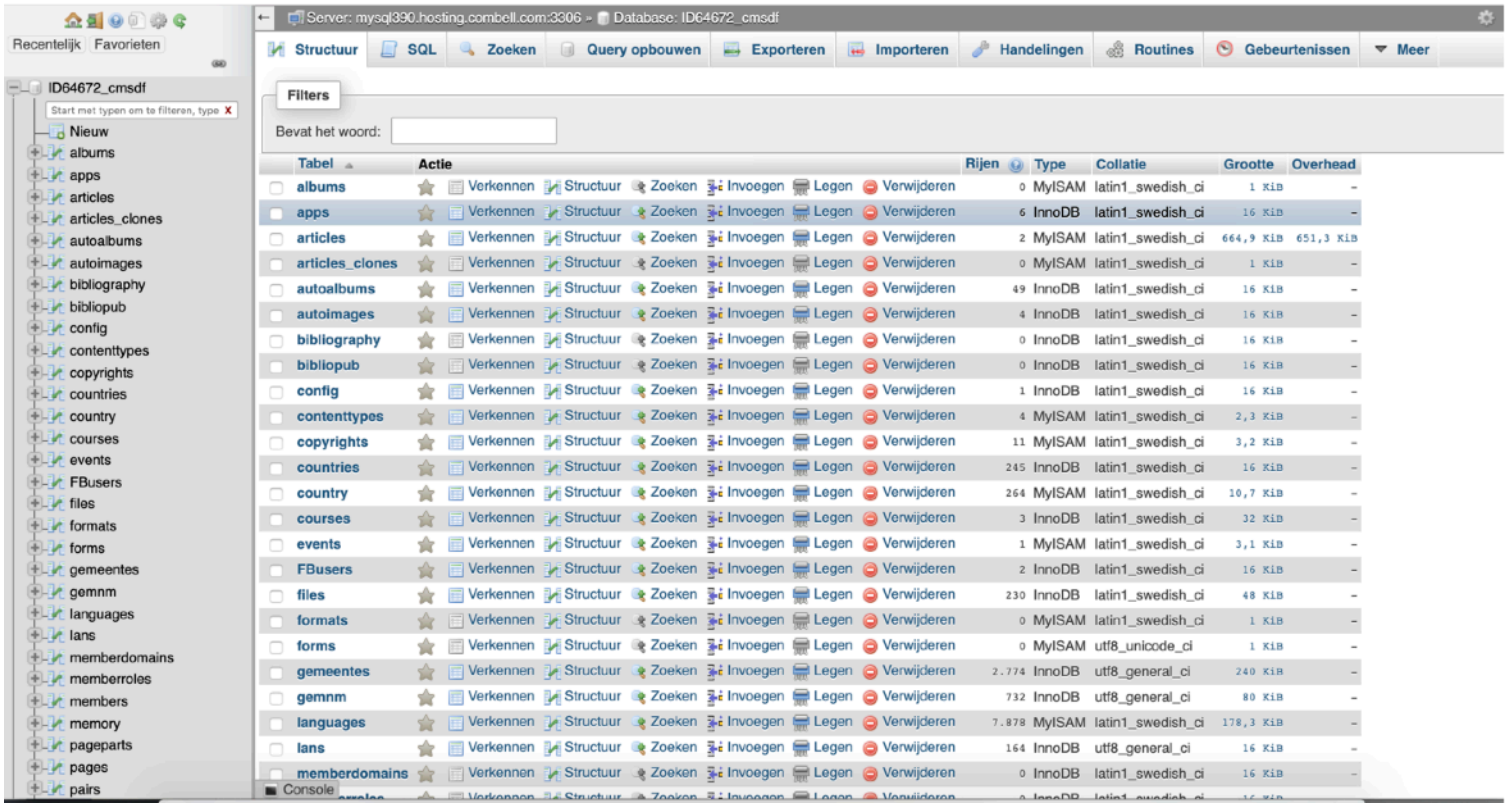
Databases of all kinds and sizes

Database software for home, garden and kitchen use, like MS Access and Filemaker Pro, offer a user-friendly interface for CRUD commands. However, MS Access databases can also be used for data storage behind corporate frontends programmed in language like Visual Basic or Visual C++.

Of course, large applications that have to able to process a great deal of data require database systems to match. For websites, the database system MySQL is the undisputed No. 1. It is open source and free to use. Very large applications like ERP software may turn to MySQL or ORACLE DATABASE.

Databases can also collect information automatically. In a number of cases this raises serious questions about privacy. The USA, UK, Canada, Australia and New Zealand listen in on communication channels worldwide through the espionage network ECHELON. The marketing company Doubleclick tries to link email addresses to IP addresses. Companies and some governments find this kind of data very valuable.

Illustration: PHPMyAdmin, a frontend for MySQL



ACID to make sure

Data integrity is critical in most cases. A financial database behind bank software, for instance, can't afford to bungle a few bank transfers. The same is true for databases behind ERP or CRM software. Staff and stock management must be consistent and correct, otherwise there is a risk of running afoul of your accounting department. That is why this type of database must meet the following four requirements:

Atomic	A CRUD transaction must either succeed completely or fail completely. There is nothing in between.
Consistent	A CRUD transaction must result in a new status or keep the old status.
Isolated	Concurrent CRUD transactions cannot affect each other.
Durable	A transaction is permanent. A commit today continues to be a commit in the future.

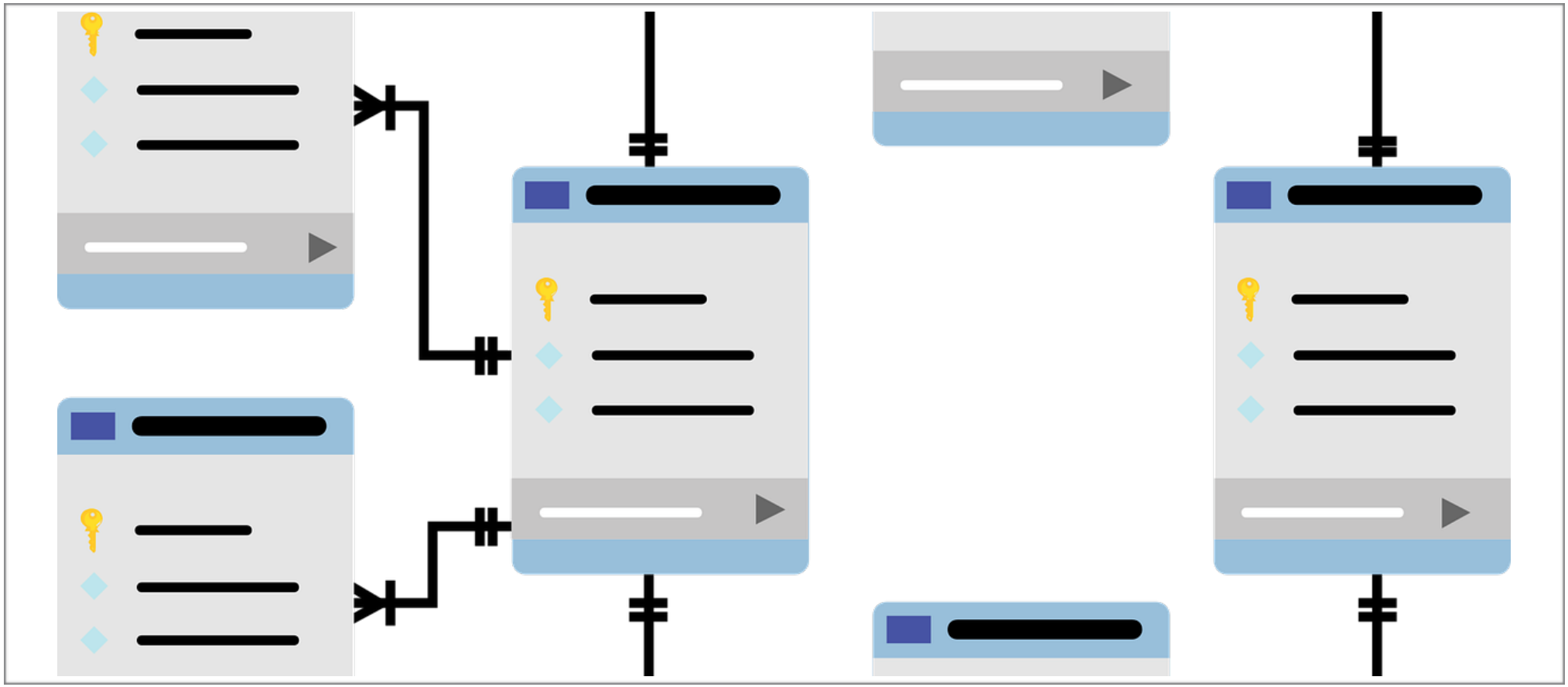
Non-relational databases

Not all databases benefit from a relational structure. After all, the fixed tables and columns structure makes it difficult to "scale" a database.

Example: suppose you were to set up a database with an overview of all ingredients in every possible food product, how can this be accomplished with a classic relational database? Or you want an analysis of clickstreams on a website. Not every website has the same number of links for a visitor to click on and not every visitor behaves in the same way. Sometimes the fixed structure of the database is not convenient and you feel you need to add more fields or columns. ACID is also not very relevant in these scenarios. Take the example of clickstreams: a few clicks more or less won't make a difference.

*For these scenarios, companies increasingly resort to so-called NoSQL databases. The "No" doesn't mean that these databases are anti-sql. NoSQL means "Not only SQL". Known examples include: Digg, Twitter (both from MySQL to Cassandra), LinkedIn (from Oracle and MySQL to Voldemort), Craigslist (from MySQL to MongoDB) and of course Facebook (from MySQL to Hbase).*1

(1. Source:https://computerworld.nl/development/74798-wat-is-nosql)



Knowing, understanding

1. Fully explain the terms RDBMS, CRUD, ACID, SQL and NoSQL.
2. What are data types? What use are they?
3. Explain the use of databases.
4. Explain the difference between relational databases and NoSQL databases.
5. Why is ACID critical?
6. What are the advantages of RDBMS?
7. Explain how a relational database can make your data more consistent and compact.

Doing

8. Come up with a meaningful database structure based on given data.
9. Give an example of data for which an RDBMS doesn't fit.

Chapter 4

COLLECTING DATA

- **What digital trails does each of us leave behind?**
- **How is data collected?**
- **How can we collect data ourselves?**

What trails do we leave behind? We email, post on social media, create files on our computer. All this digital activity leaves traces. Some we are aware of, others considerably less so.

Shopping streets are littered with cameras – smart or otherwise – that register our movements and look for suspicious behaviour. They not only register, but also store their data for possible later analysis.

Authorities install increasingly more SPECS cameras along secondary roads and highways. After all, we are all potential traffic offenders, and we are left in no doubt about that.

How is data collected? How much data do we produce in a single day? Why is the produced amount of digital data exponentially increasing?

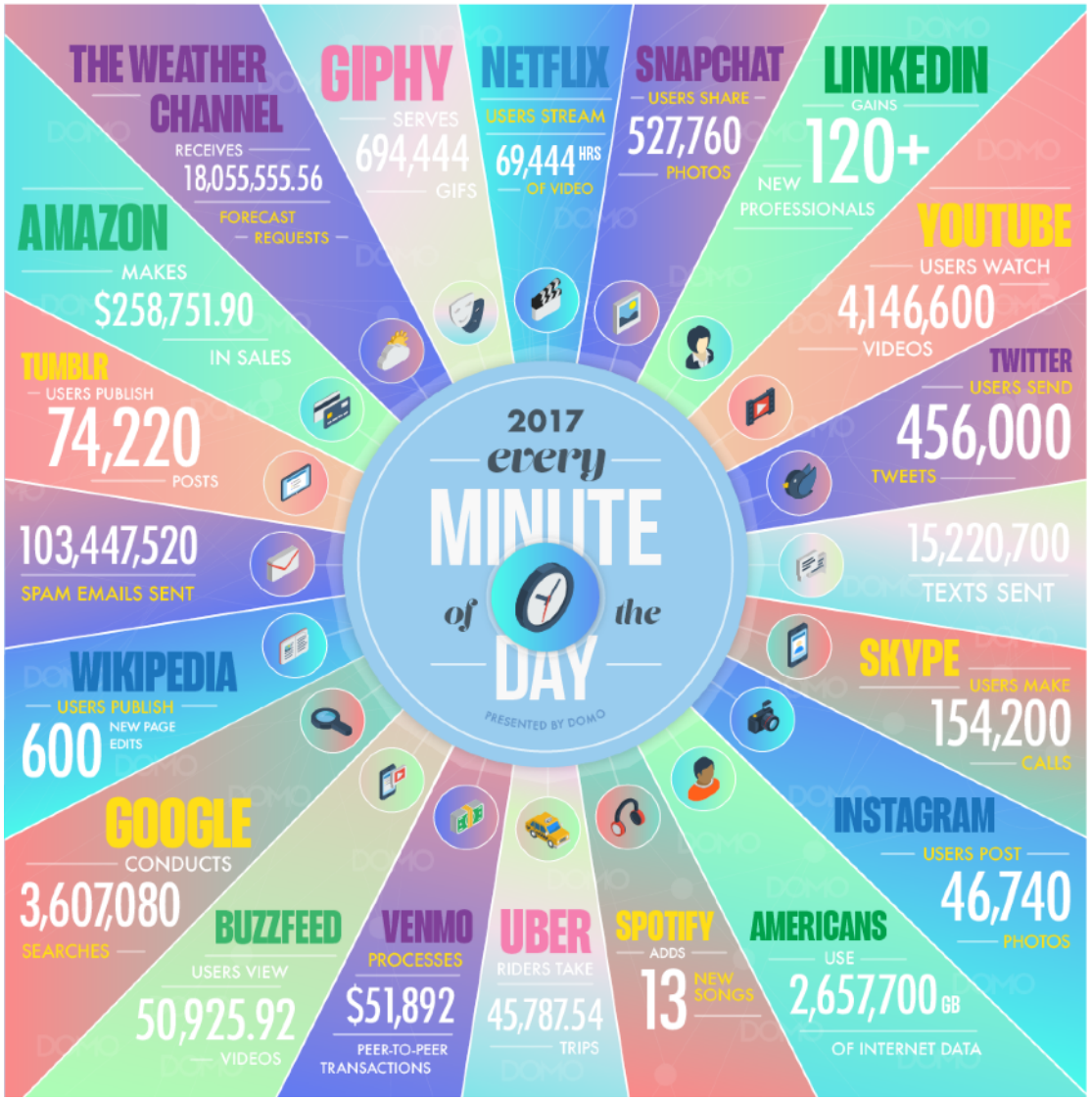
Data revolution

The amount of digital data generated by humankind on a daily basis is staggering. The rise of the Internet of Things is even accelerating this further at an unprecedented pace.

The Internet of Things – smart devices hooked up to the internet – collects massive amounts of data. In 2006, there were about 2 billion devices. Right now we're looking at 200 billion devices with numbers rising every second.

Ninety percent of all available data in the world has been generated in the last few years.

The smart revolution gave us the IoT, smartphones, digital assistants (Google Home, Amazon Alexa, etc.), smartwatches, beacons, smart cars, smart homes, and so on and they all contribute to the generation of data.



"By 2025, it's estimated that 463 exabytes of data will be created each day globally – that's the equivalent of 212,765,957 DVDs per day!"

(Bron: https://www.visualcapitalist.com/how-much-data-is-generated-each-day/)

Causes of growth and purpose

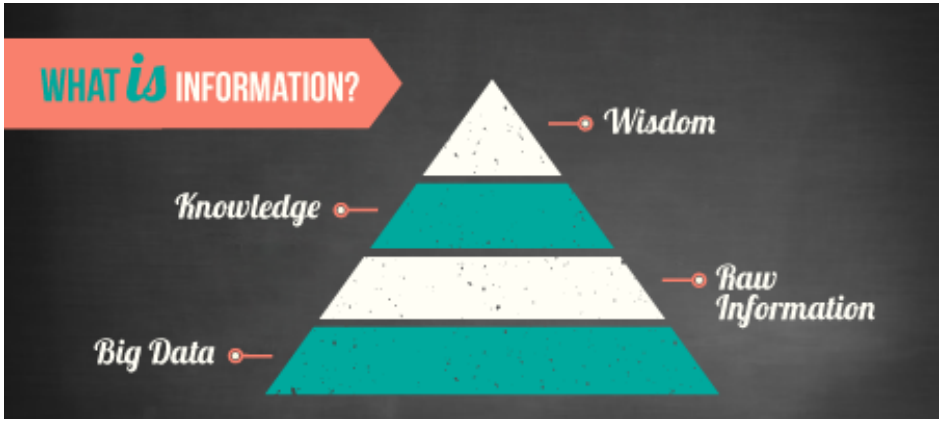
Data isn't always produced intentionally or purposefully. Often new technologies contribute to an increase in data production. At the same time, data in and of itself has no purpose.

Data production and technology seem akin to flowers and bees, spurring each other on. The miniaturisation of storage media (hard drives, SSD, etc.) enables increasingly more data to be collected and stored. The increasing processing power of computer systems allows great quantities of data to be analysed. Concepts from artificial intelligence like neural networks had already been conceived in the late fifties of the last century (John Von Neumann) but are only now becoming usable.

Data can also be collected purposefully, either by storing new data or sifting through existing data. The purpose of such data can be very diverse:

- Scientific
- Academic
- Marketing
- Predictive
- Political
- Legal
- etc.

It is clear that the Chinese government uses data collection and analysis for different purposes than most universities. Purposeful data collection leads to data analysis. Analysis leads to accumulation of information and 'knowledge'.

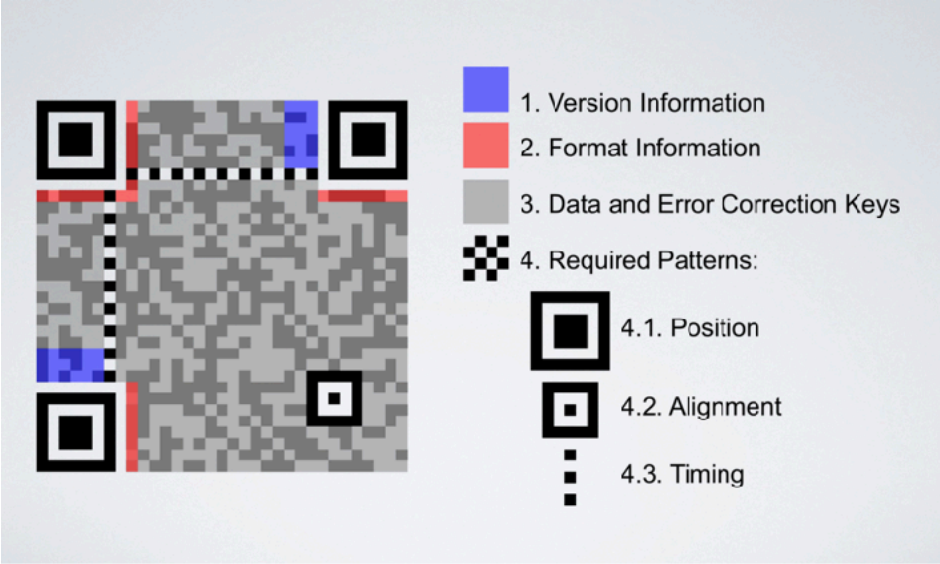


Knowing, understanding

1. Why is the amount of data growing exponentially?
2. List some of the main reasons behind the burgeoning amount of data.
3. What are the principal reasons for data collection?

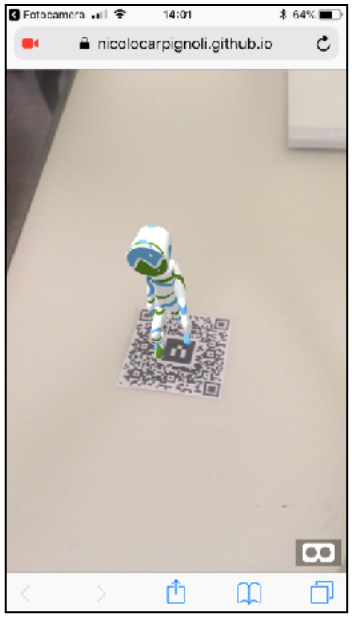
Doing

1. Give examples of the purpose and the use of data collection.
2. Locate and open a dataset with Dbeaver.



Augmented reality adds digital information to the analogue reality. Digital data appears in our natural environment. This kind of immersive interface will become the standard for operating computer systems.

(Source: https://medium.com/chialab-open-source/how-to-deliver-ar-on-the-web-only-with-a-qr-code-e24b7b61f8cb)



How do we exchange data?

The advent of the smartphone has set up almost everyone with a digital pocket assistant. Smartphones are typically equipped with a whole slew of **sensors** (GPS, accelerometer, gyroscope, etc.) and communication technologies (4G, Bluetooth, NFC, etc.) that **enable the exchange of data without direct human intervention.**

Quick response

The camera on your smartphone also doubles as a scanner for documents. The right app can then convert these scans into editable text again using **OCR (Optical Character Recognition).** Many smartphones recognise QR codes automatically already. The QR (Quick Response) code seems to have become the successor to the barcode. It can contain a price, but also text information or a link. Banks use it in their mobile apps to approve payments. When a publisher adds a QR code to a book, the reader can use the referring link to get access to more online information about it.

RFID-tags.

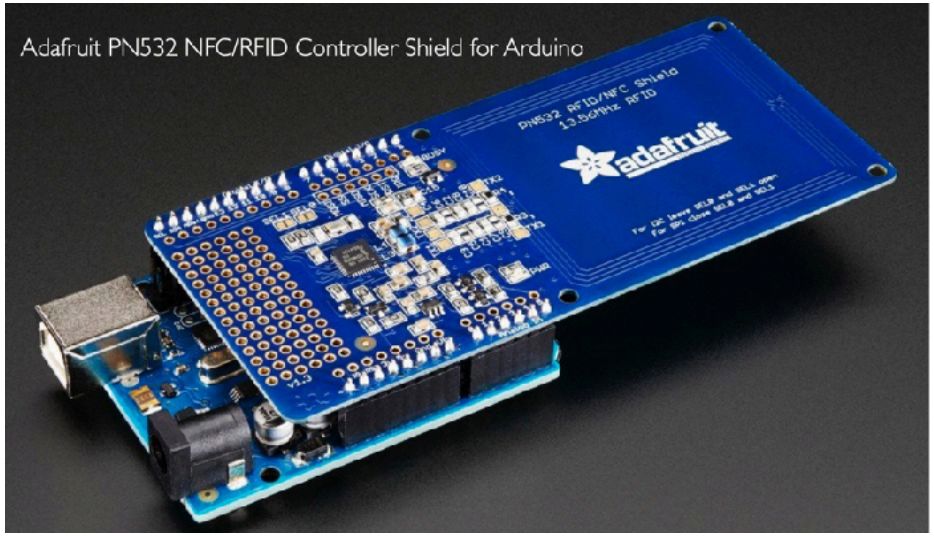
QR-code

QR is a technique that is relatively easy to integrate. You need a piece of software to generate codes. End users need a camera app on their smartphone as well as the necessary software to read the code.

Smart tags: RFID, NFC

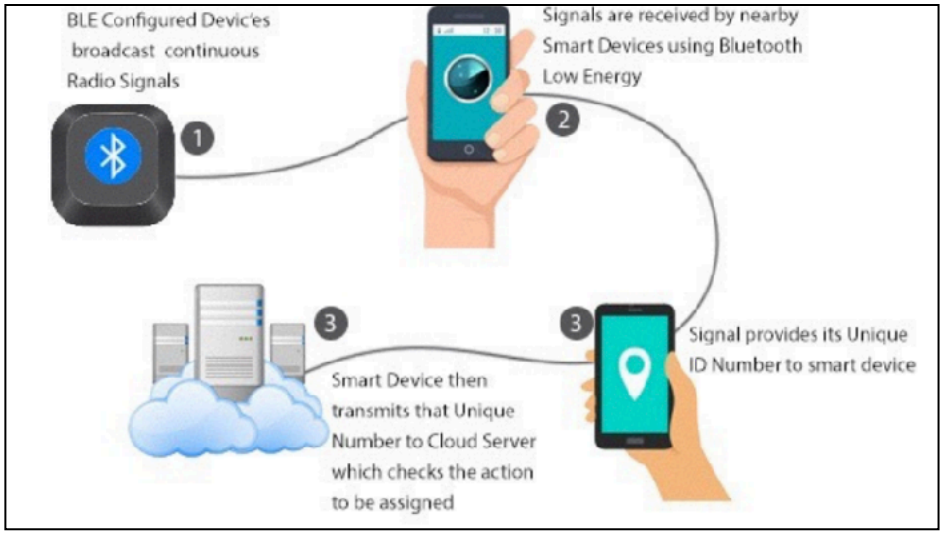
RFID, NFC and BLE use radio waves to exchange information.

RFID (Radio Frequency Identification) is a technique to uniquely identify physical objects and even living beings. A complete RFID system is composed of **a tag, a reader and an antenna**. The reader sends out a signal via the antenna. The tag subsequently responds with unique information (e.g. a product code, a price, etc.). Active RFID tags even have their own power source and can transmit radio signals as far as 100m. Passive RFID tags receive their energy from the electromagnetic waves transmitted by the RFID reader. Since the radio waves must be sufficiently strong, passive tags only work well within a limited range (direct contact about 25m).



NFC (Near Field Communication) is a specialised member of the RFID technology family. An NFC device can be both reader and tag. NFC devices have to be very near each other, no more than a couple of centimetres. That's why shops like to use it for contactless payments. You simply need to hold your bankcard against the terminal. Safe exchange of information is the main purpose of NFC.

NFC has been finding its way to a new generation of smartphones. If you hold two NFC smartphones near each other, they can exchange pictures (e.g. by swiping). But it is also possible to add an NFC tag to posters to exchange information with the smartphones of consumers. Audio equipment with NFC capability can play music by simply putting your smartphone on it. In short: NFC (Near Field Communication) is a wireless way to exchange small amounts of data within a 10 cm radius. The chip in your smartphone communicates with other NFC devices like payment systems, speakers or other smartphones.



Indoor positioning

Beacons or iBeacons are small Bluetooth transmitters (Bluetooth uses radio waves). A beacon looks like a small box and contains a **Bluetooth Low Energy (BLE)** chip and a battery. Every beacon continuously sends out a unique number. A smartphone with BLE support is able to receive this signal. If you know which beacon is located where, you can determine someone's location within the premises of a building or shop. Beacons, therefore, are ideal for indoor positioning since GPS is of limited use there.

Apple embraces BLE as an alternative to NFC. Whereas NFC only works at distances of centimetres, BLE reaches as far as 70 meters. One significant advantage is that it doesn't involve a complicated installation. They can be put anywhere. An iOS or Android device knows who you are. In combination with the beacons, the smartphone app therefore knows where you are. If we permit ourselves briefly to look beyond the concerns for privacy, this technology offers many **new possibilities**: *indoor positioning and navigation, sales promotion (in-store promotion), loyalty (loyalty cards), access control (alternative to access passes), transactions (contactless payment), and so on.*

(Bron: https://www.konstantinfo.com/blog/wearables-become-desirables/)

Wearables

Wearables are mostly associated with health hypes. A wearable is a gadget that is carried on the body and collects data about your body. Most of these devices collect personal data about the wearer. They dispense advice about health in general and you specifi cally. Activity trackers are clever devices that keep track of how much you move and how many calories you burn. Sport watches have additional functions like heart rate and sleep rhythm monitoring.

Of course manufacturers are interested mostly in the data they can collect about users. It is part of big data collected by large companies like Google and Apple which they sell to insurance companies and other clients.

The **LilyPad Arduino** is a flexible microcontroller that makes your clothes or textiles smart (e-textiles). The developers themselves refer to it as **softwear**.

IoT and miniaturisation

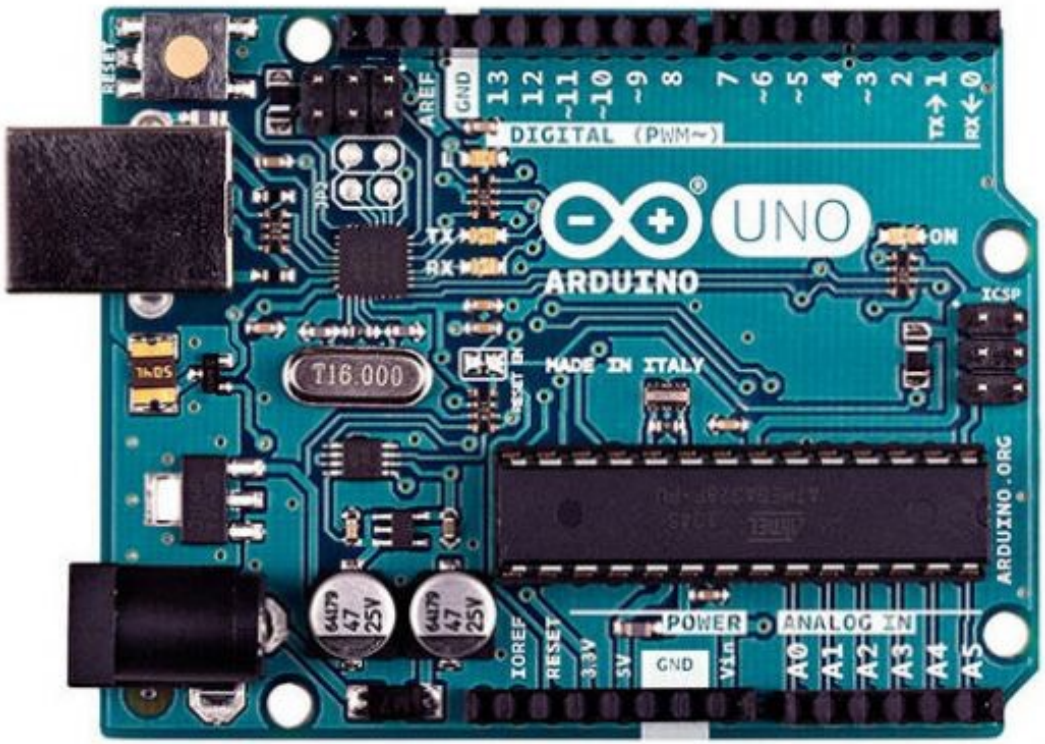
Our computers, laptops, tablets and smartphones are not the only devices in our environment that collect data about each and everyone of us.

There are other implementations that are also referred to as computers for convenience sake, but don't really fit the definition. A microcontroller is one such 'computer'. **Microcontrollers** or **microprocessors** are used to control electronic equipment.

Many modern devices contain such microcontrollers: magnetrons, cars, laundry machines, some telephones, and so on. They often contain sensors that collect data from their environment. Often they can be controlled through the internet (so remotely).

Embedded systems and microcontrollers are taking the world by storm. We have briefly touched upon the phenomenon of the 'the internet of things' already. By and large it refers to the fact that gradually, every domestic device, but also cars, surveillance cameras and home automation systems have embedded systems on board. When all these things are hooked up to the internet, the result is an all-encompassing network of digitally connected devices: the **Internet of Things** (IoT). We will not be asking questions about the consequences and dangers for privacy and ethics here.

In short, embedded systems and microcontrollers have become an intricate part of today's world.



Arduino prototyping board with microcontroller

Embedded systems offer many **advantages** compared to a 'computer'.

1. Processing is in real time (imagine your car working on Windows)

2.Fast

3.Low energy consumption

4.Embedded systems are much cheaper.

One **disadvantage** is that the functionality is very limited. You can't put a word processor in your refrigerator or video editing software in your car.

The **Arduino** is an open source development platform with microcontroller. It allows fast prototyping of digital/electronic devices.

Illustration:

The Leuven-based IT company EASICS developed image recognition hardware for the sorting machines of the Belgian company BEST. The EASICS hardware is able to recognise – at astonishing speed – 'undesirable' elements amongst fries, currants, shrimp, nails or anything else speeding by and then remove that element with a targeted jet of pressurised air. A 'multifunctional' computer, no matter how fast and powerful, would not succeed in carrying out the task so quickly. Because the programme code is hard coded, processing is astonishingly fast.

Another term to pop up frequently in addition to 'microcontroller' and 'embedded system' is **'system on a chip' (SoC)**. A SoC integrates all components of a computer and/or electronic system on a single chip.

SoCs are used extensively in the market of consumer electronics. An embedded system often has a SoC at its core.

Illustration: Various SoCs in a smartphone.

Knowing, understanding

1. Name at least four recent technologies for wireless data exchange and explain each one.
2. Explain QR, RFID, NFC, BLE, Beacons, wearables, IoT, microcontrollers, SoC.
3. How can data on visitors be collected inside a building?
4. What effect do these technologies have on the exponential growth of data?

Doing

5. Come up with situations in which these technologies could have a meaningful use for data collection.

Collecting data

Data can be collected from a massive number of sources. However, the multitude of data doesn't make it any easier to extract meaningful information. As stated above, data collection in and of itself serves no purpose. The data needs to be analysed and conclusions drawn from it.

Sometimes, the data you require already exists. A collection of data (like the one fed into AI algorithms as a training set, see Chapter 6) is often referred to as a **dataset**.

It seems fairly straightforward to look for your information on the internet or the world wide web. Almost all webpages are built using the HTML language which, although it does contain structured information, is often found lacking in terms of semantics.

In addition, the web has many sources of information that can be accessed differently by means of a **REST API**.

HTML and online data

We all visit websites and use social media and applications on the world wide web. The content we are being served is a combination of **HTML** (content, structure, links), CSS (markup, animations) and Javascript (interaction). These three 'languages' are the building blocks of what is being shown in the browser.

However, a great chunk of the information and content behind all these websites and web services is being kept in online databases (very often MySQL).

Using backend technologies like PHP, Node.JS, Java, and so on, these database records are magicked into webpages.

REST API

A number of larger online platforms also offer other output formats beyond HTML. For instance, Wikipedia data is also available as JSON data. This format is particularly handy when you want to develop your own software with the intention of using data from other websites. In some cases this is for free, in other cases such service requires payment. After payment you will receive an authentication key that has to be embedded in your programming code.

In order to query data, it's often enough to set up the URL with a correct syntax. The technique to query information in standardised and/or structured formats is called **REST (representational state transfer).** The services rendered and the way to query the data is described in an **API (application programming interface)**.

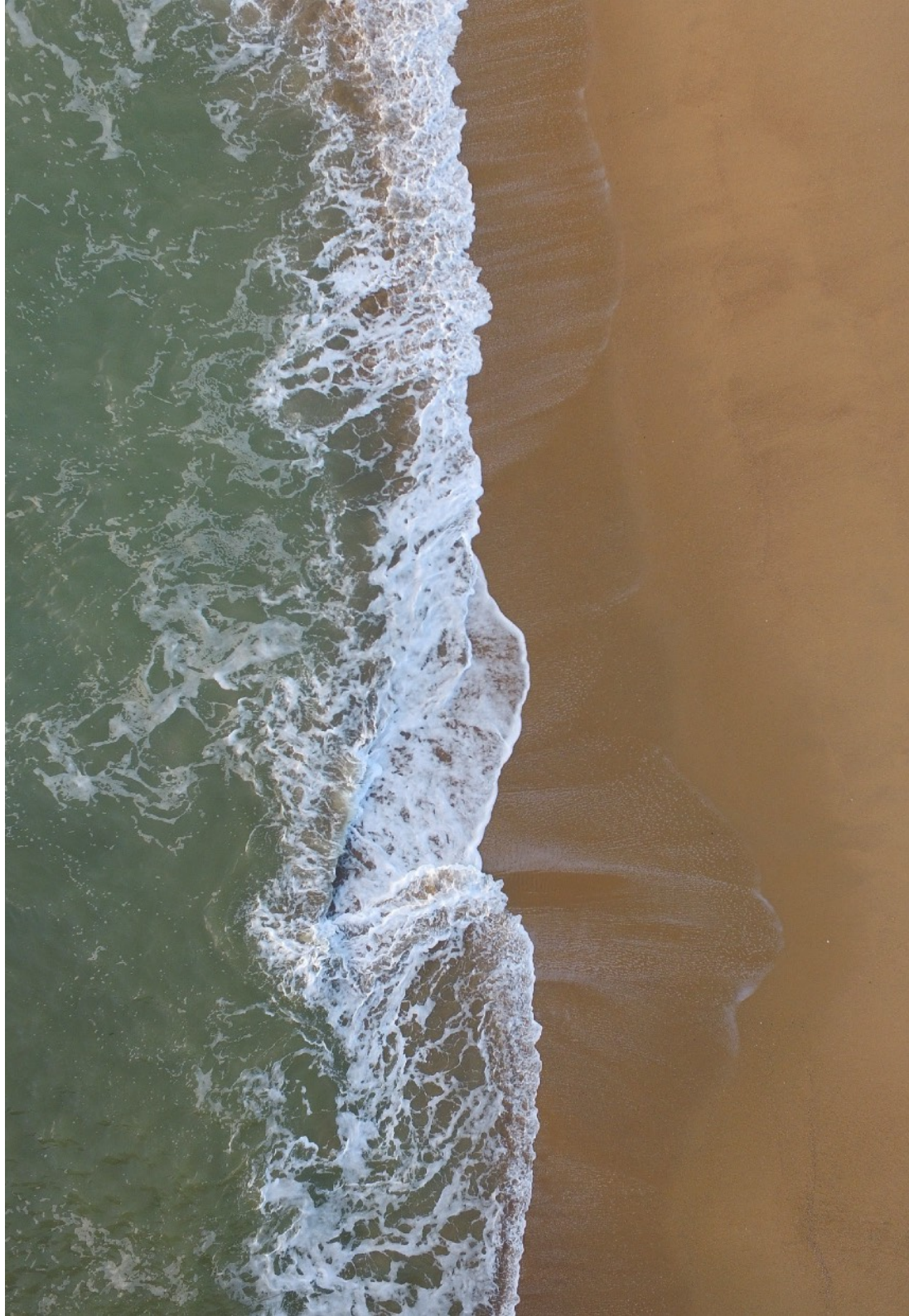
Webscraping

Web scraping is a technique used to automatically copy the content of webpages or REST APIs to a computer or to another website of web service. Generally, this requires setting to work with programming code.

Data cleansing

Data cleansing is the process of detecting and correcting (or removing) imprecise or otherwise flawed records from a record set, table, database or dataset. After cleansing, a dataset must be consistent with comparable datasets in the system. The detected and possibly removed inconsistencies may originally have been caused by user input, transmission or storage errors.

The validation process can be strict (like refusing any address without a valid postal code) or vague (“fuzzy”, like correcting records that are a partial match with existing, known records).



Knowing, understanding

1. What are datasets?
2. What is a REST API?
3. Explain the use and purpose of web scraping.
4. What techniques allow for automated data collection from online sources?
5. What do you understand by ‘data cleansing’?

Doing

6. Being able to offer tips to cleanse data.
7. Being able to scrape specific information from the web as an assignment (during class).
8. Being able to scrape specific information from web services using REST APIs as an assignment (during class).

Chapter 5

FROM ALGORITHM TO SMART COMPUTER

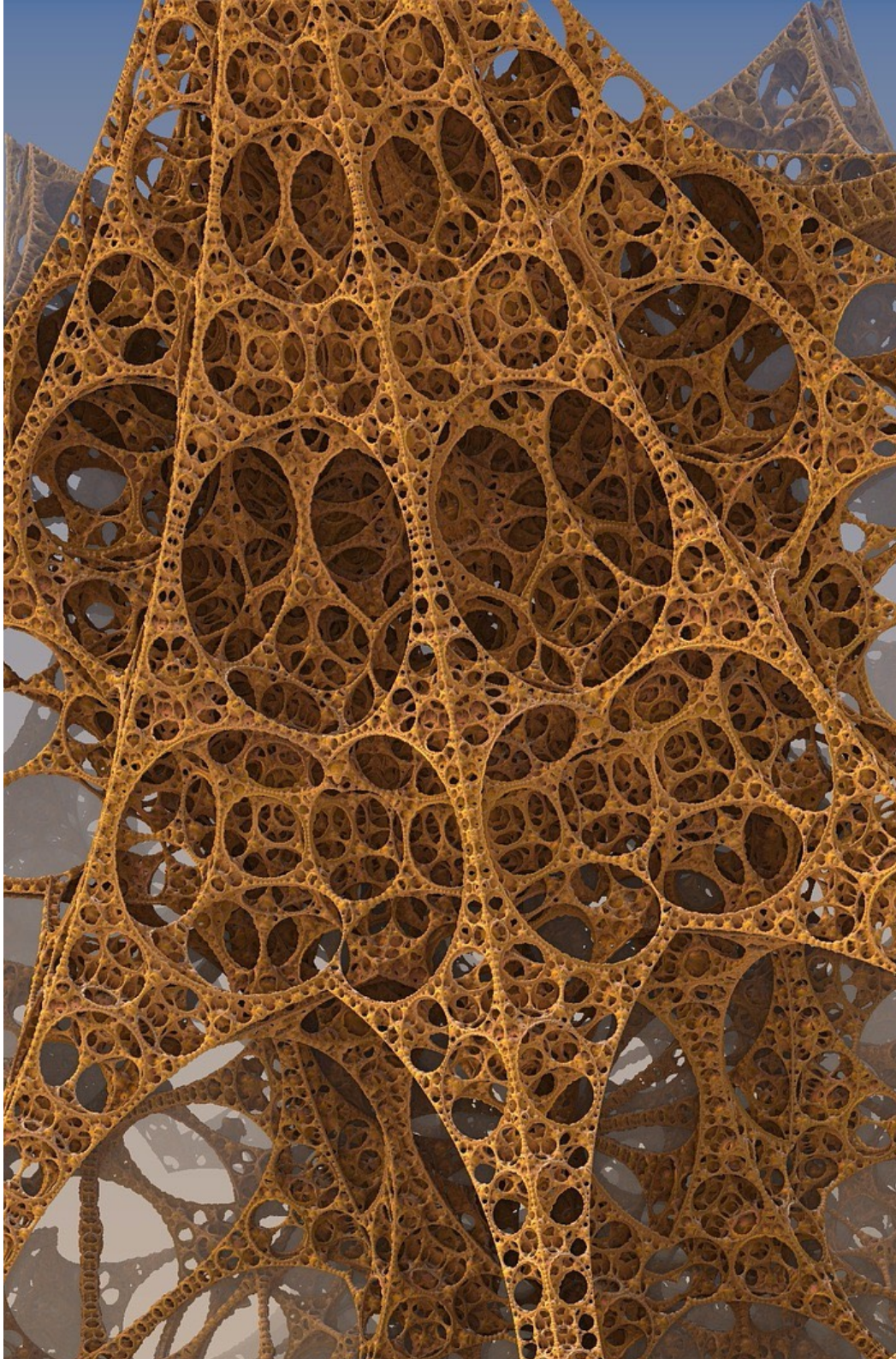
- What are algorithms?
- Why do algorithms play such an important role in today's world?
- What is automation?

Algorithms

You are likely to have come across the term algorithm before. It has nothing to do with rhythmic music even though Shazam does use algorithms for pattern recognition in music. Google uses the PageRank search algorithm and MP3 music files owe their relatively small size to a compression algorithm saving space on your media carrier. Algorithms play a crucial role in big data processing.

Algorithms are omnipresent in the digital world, but what are they?

Algorithms in this case are bits of programming code which allow a computer to carry out certain tasks in the correct order. Below are the most important kinds of software algorithms.



Dijkstra: de shortest path

The Dijkstra algorithm is known as the Shortest Path First algorithm. It finds the shortest path between two points.

'The Dijkstra algorithm is used in various internet protocols for finding the shortest route between computers or routers. The Dijkstra algorithm is guaranteed to come up with the shortest path between A and B, if indeed one exists, but may take long to get there. The speed of the algorithm depends on the way in which the nodes and lines are stored in the computer's memory.'

(Source; GUNNINK, M., "Padvinder vindt pad: over Pathfinding", (http://kninnug.nl/padvinder/index.html), Geraadpleegd op 20 oktober 2015.)

Rolling dice

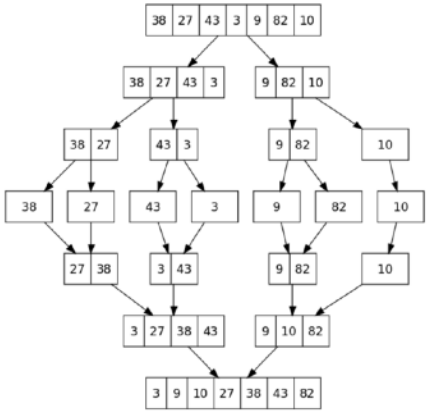
Playing dice or playing the lotto is literally playing with "chance". A computer can likewise generate random numbers.

The random number generator function is an easy function to call up in most programming languages. Random number generators are important in security and cryptography, games, etc.

Sorting algorithms

At school, you learn how to sort words alphabetically. The first step is to consider the first letter. Words starting with the same letter are then sorted by looking at the second letter and so on.

Sorting thus requires an algorithm, a strictly phased plan to arrive at the desired outcome (alphabetical order). A sorting algorithm arranges the elements of a list in a particular order: alphabetical, descending, ascending, and so on. For instance, when you sort numbers by size in MS Excel, what you are actually doing is using a sorting algorithm.



Arranging words alphabetically and sorting according to number size seems very straightforward to us but it does require a phased approach even if it no longer requires much thought after a while. The better the algorithm, the quicker the procedure. Computer science has been using these kinds of algorithms since the early days of computing. John von Neumann came up with the merge-sort algorithm in 1945. It sorts data by taking it apart, sorting it and putting it back together (merging). Tony Hoare came up with the quicksort algorithm in 1959 which, as the name implies, works faster than merge sort.

In 1964, J.W.J. Williams developed the heapsort algorithm. Without these sorting algorithms, many modern computing techniques, like data mining, AI, link analysis, and so on, would be unthinkable.

Link analysis

The results that appear on top in Google search are determined by the PageRank algorithm. Facebook shows a different news feed for every user based on the 'friends' you are in touch with most (on Facebook in any case), on posts that got the greatest number of likes, etc. So what is shown you, depends on a great number of factors and is determined by a link analysis algorithm. Of course the algorithms vary, but the foundation for link analysis was established in 1976 by Gabriel Pinski and Francis Narin.

"Who uses this algorithm? Google in its Page Rank, Facebook when it shows you your news feed (this is the reason why Facebook news feed is not an algorithm but the result of one), Google+ and Facebook friend suggestion, LinkedIn suggestions for jobs and contacts, Netflix and Hulu for movies, YouTube for videos, etc. Each one has a different objective and different parameters, but the math behind each remains the same. Finally, I'd like to say that even thought it seems like Google was the first company to work with this type of algorithms, in 1996 (two years before Google) a little search engine called "RankDex" , founded by Robin Li, was already using this idea for page ranking. Finally Massimo Marchiori, the founder of "HyperSearch", used an algorithm of page rank based on the relations between single pages. (The two founders are mentioned in the patents of Google)."

Source: OTERO, M., "The real 10 algorithms that dominate our world" (https://medium.com/@_marcos_otero/the-real-10-algorithms-that-dominateour-world-e95fa9f16c04), 2014, Geraadpleegd op 19 oktober 2015.

Suggestions

When entering the same search term in Google but on two different computers by two different 'people', the search results are often different. Indeed, Googles **PageRank** sorts the search results and the Google **Adwords** algorithm offers advertisements based on your search history and profile.

While entering a search term, Google **Suggest** already offers a list of possibilities from which a selection can be made. Likewise, Amazon and Netflix show products you are likely interested in based on your search history and previous purchases. Eli Pariser refers to this as 'information determinism'. The user is stuck in a kind of filter bubble over which he or she is not really in control. So you no longer determine what you see, the algorithms determine this for you. The question is, purely from a marketing point of view, whether or not you lose potential buyers of certain products this way.

Online dating is another example of a service that is in great demand and is based on suggestion algorithms. The matching algorithm of OK Cupid (an online dating service) was provided by Harvard mathematician Christian Rudder. It makes a rough match based on shared interests. The question is, of course, if someone with similar interests is indeed the best match. Which is why the algorithm also considers how much weight the other party attributes to each question.

Security

An increasing number of computer users tape off their webcam because they are aware that internet adventures may be marred by privacy violations. And yet we count on security when we do online banking, book a trip on Booking.com or enter our credit card data in an app store or PayPal. Without this sense of security you would be crazy to leave your bank information anywhere.

The **RSA** algorithm (written by the company RSA) and the **Secure Hash algorithms** ensure that our data is exchanged securely between our home computer and the servers of banks and online shops by encrypting it. By extension, **integer factorisation algorithms** play an important role in the world of cryptography as well.

Error correction and autotuning

Proportional Integral Derivative Algorithm. It certainly sounds impressive but what it does is equally brilliant. A **PID controller** continuously calculates error values as the difference between the measured signal value and the desired result and makes appropriate corrections. PID controllers are used in the industry but also in more popular applications. Singers are not always pitch perfect in the recording studio. As the recording progresses they are often just above or just below the right note which is not to say they are entirely off-key, they are just rarely dead on.

Autotuning makes sure all these little errors disappear out of the final mix. The song Believe by Cher is said to have been the first pop song to use autotuning. You notice the difference immediately when comparing recordings from the sixties or seventies with those of the digital era starting in 1984. When you have a chance listen to the Communique album by the Dire Straits and the pitch perfect digital sound of Brothers in Arms by the same band. Not that we accuse the Dire Straits and Mark Knopfler of autotuning, but the digitization algorithms have had a lasting effect on all recordings since the arrival of the CD.

Conversion and compression

The **Fourier transformation algorithm** and the **fast Fourier transform** or FFT algorithm are what compression algorithms like JPG and MP3 are based on. A Fourier transformation turns a sound signal or light intensity into a wave function.

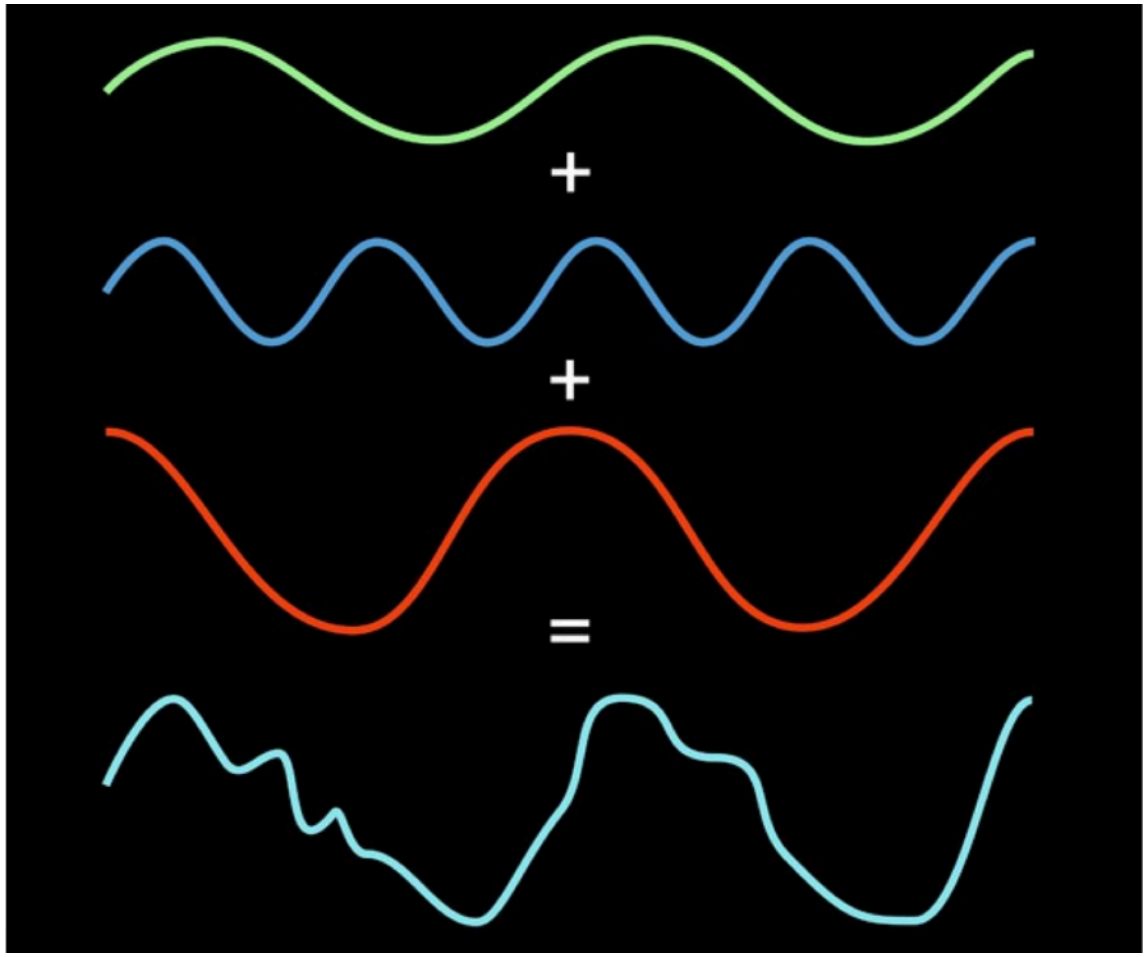
The internet, Wi-Fi, smartphones, computers, routers, communication satellites or just about any device that incorporates some sort of a computer, uses these algorithms. It measures analogue signals against time and represents them as waves based on the frequencies measured. Turning signals into (wave) functions also allow the intensity of waves to be transformed.

The FFT algorithm, for instance, is used in image filters that transform the intensity of colours or lighting. With compression, small differences in intensity are discarded thereby significantly reducing file size.

“Fourier transformations are used to turn analogue signals into digital data, for instance in signal processing, but they also play a role in compressing images. FFT is used to reduce a complex signal into a number of components that are easy to represent digitally. Which is to say an analogue signal is divided into little pieces which are analysed and converted to a digital signal.”

Source: DE MOOR, W., "Sneller algoritme voor Fourier-transformaties ontwikkeld", (http://tweakers.net/nieuws/79444/sneller-algoritme-voor-fouriertransformaties-ontwikkeld.html), 2012, Geraadpleegd op 19 oktober 2015.

Compression algorithms play a particularly important role. They ensure that data becomes smaller, but often this is accompanied by a loss of quality. The algorithms make a trade-off between quality and quantity.



Surveillance and projection

In 1948, George Orwell wrote the book 1984 in which he heavily criticised the USSR.

He depicted an “inhumane dictatorial one-party state that regulated every realm of life. The omnipresent leader of the Party and the country was called Big Brother. Every inhabitant of the country was continuously monitored by cameras including when indoors. Thus the catchphrase ‘Big Brother is watching you’ was born.”

Meanwhile in the West, we may not be living in dictatorial one-party states, but Big Brother does seem very real. Every one of us is monitored, albeit not by people but by algorithms. From the USA, the National Security Agency (NSA) and its international partners (the Five Eyes: USA, Australia, Canada, New Zealand, United Kingdom) survey millions of people around the globe. They monitor telephone conversations, text messages, emails, webcam footage, GPS locations, etc. The quantity of collected data is far too great to be analysed and interpreted by people. The analysis is carried out automatically by means of powerful algorithms.

Some algorithms like IBM’s CRUSH even take it one step further. CRUSH or Criminal Reduction Utilizing Statistical History provides a predictive analysis which is, in the main, intended to prevent crimes. The Memphis police service was able to reduce crime numbers by 30% thanks to CRUSH. Compared to 2006, the number of violent crimes has been reduced by 15%. Based on statistical data, data aggregation and algorithms, the software maps out criminal hot spots.

This allows police units to act proactively and arrive in a manner of speaking before the crime has been committed. In the future, it will be possible to trace criminals very quickly based on internet activity, GPS and biosignatures, and suspicious behaviour.

“In the future, these systems will largely take over the work of analysts. Criminals will be tracked by sophisticated algorithms that monitor internet activity, GPS, personal digital assistants, biosignatures, and all communications in real time. Unmanned aerial vehicles will increasingly be used to track potential offenders to predict intent through their body movements and other visual clues.”

Source: DVORSKY, G., "The 10 algorithms that dominate our world", (http://io9.com/the-10-algorithms-that-dominate-our-world-1580110464), 2014, Geraadpleegd op 20 oktober 2015.

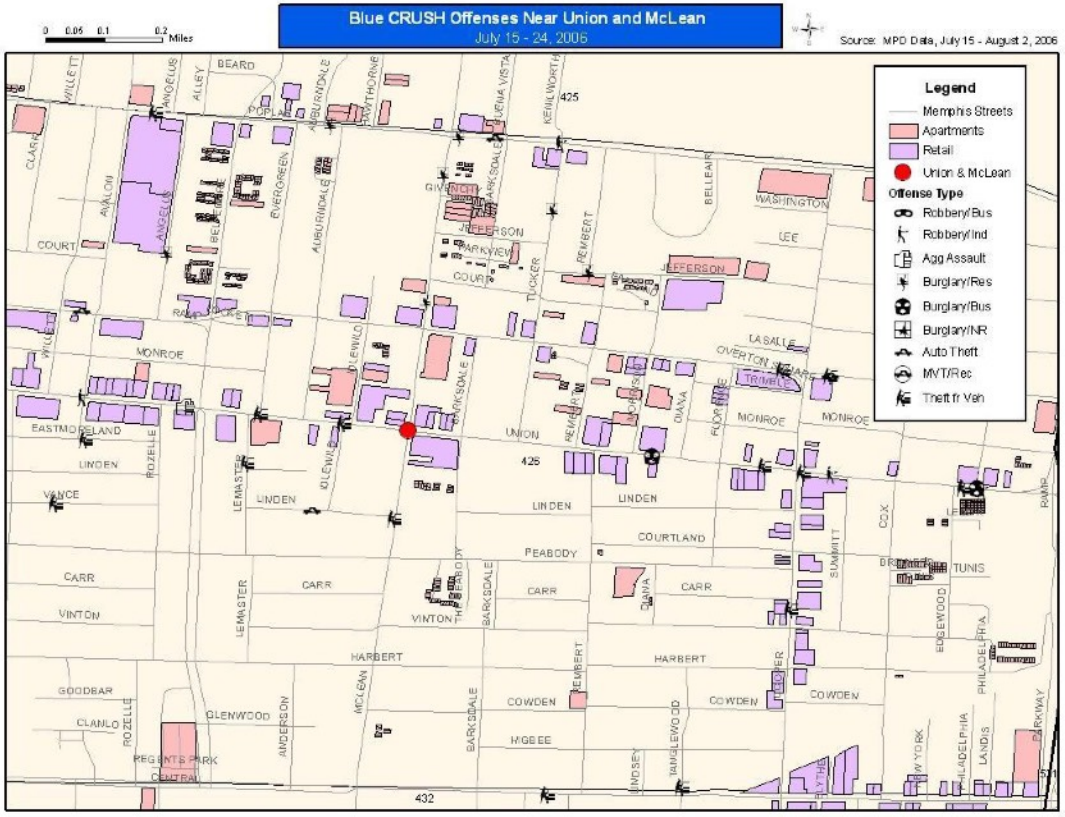
The 2002 Steven Spielberg film Minority Report already gave us a peek at the future this could lead to.

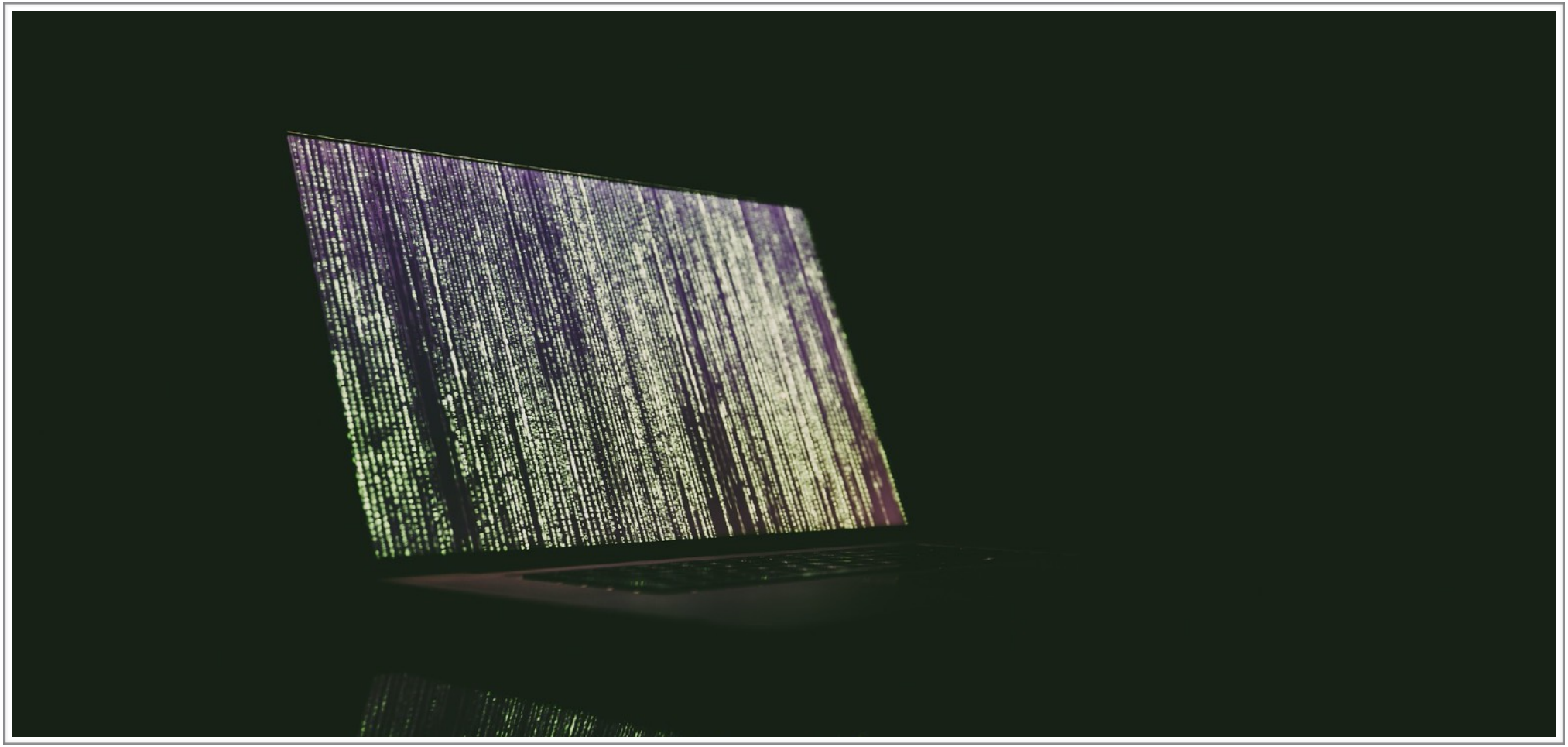
Algorithms for predictive analysis have long been in use in the world of the stock exchange. The analysis of transactions whipping past at ludicrous speeds is carried out by smart algorithms. Sometimes the projections turn out to be incorrect, like the 2010 Flash Crash.

The advent of AI and smart cameras makes it even easier to monitor and survey citizens.

In China, Big Brother is no longer fiction:*"This fictional scenario is now a daily reality in the People's Republic of China, thanks to a massive police surveillance apparatus powered by Big Data and artificial intelligence. For example, trains now require national ID's to buy tickets, which allows the government to block human rights activists or anti-corruption journalists from traveling. In Xinjiang province, home of China's Uighur Moslem minority, the government uses AI-sifted Big Data to scrutinize anyone entering a mosque or even a shopping mall thanks to thousands of checkpoints requiring a national ID check-in—and which can collate real-time data with other personal information on everything from bank accounts to family planning."*

Source: HERMAN, A. "China's Brave New World Of AI", (https://www.forbes.com/sites/arthurherman/2018/08/30/chinas-brave-new-world-of-ai/), 2018, Geraadpleegd op 5 april 2019





Knowing, understanding

1. Explain what an algorithm is.
2. Name at least five important algorithms and explain.
3. Offer examples of practical applications for the algorithms you have named.
4. Explain why algorithms are extremely important in data processing.

Doing

5. Given a dataset, identify algorithms that could help processing it.

The evolution of automation

Many people fear that the advent of artificial intelligence will lead to an enormous loss of jobs. How will our children and grandchildren be able to earn an income? Why do people make machines and technology that make them redundant?

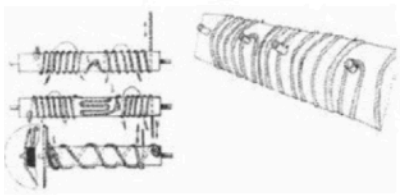
How are machines evolving? First people developed tools to make light work, requiring less muscle. After all, it is easier to hammer down a nail with an actual hammer rather than with your bare fist.

In other words, human kind has developed tools that lighten physical work but that has also lightened mental work and maintenance. The invention of script, for instance, meant you no longer had to commit everything to memory. Using draught animals, windmills and watermills meant a lot of human muscle power that could be spent elsewhere.

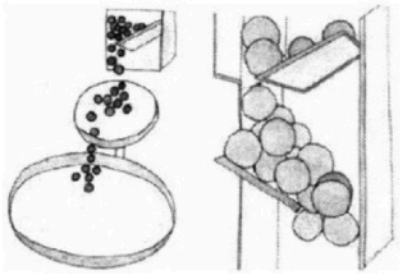
Yet generally tasks still required human supervision. You still needed to learn how to complete tasks successfully, how to wield the tools. The various steps in a task, job or function still had to be taught.

Indeed, every tasks consists of a number of steps, like a recipe. In math and computer sciences this process is referred to as an **algorithm**. As long as 2,000 years ago, humankind was already inventing machines into which such algorithms were programmed. A clock is a good example of this. It contains a complex mechanism that allows the passage of time to be tracked. One of the best known examples of automation is the way cars are built on a factory's assembly line.

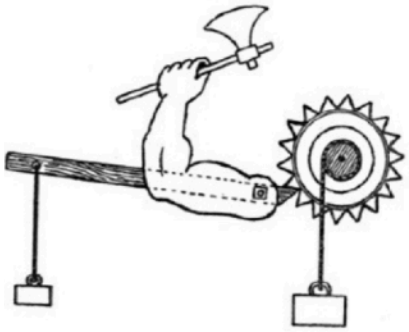
Pictures: The Computer of Antikythera, 2th century BC.

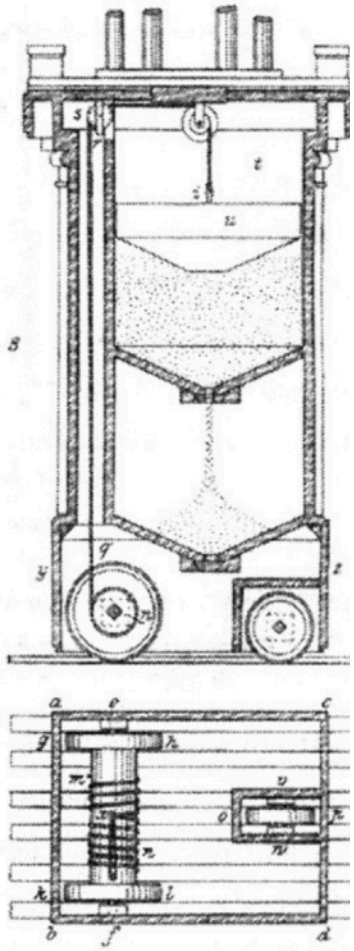


De touwen en de pinnen bepalen de beweging van de assen. (Bron: Manuela Rausch, *Die Programmgesteuerten Automaten Des Heron Von Alexandria*, 2012)



Geluidseffecten in het automatische theater. (Bron: Manuela Rausch, *Die Programmgesteuerten Automaten Des Heron Von Alexandria*, 2012)





(Bron: Beck T., *Herons des älteren Automatentheater*, Jahrbuch des Vereines deutscher Ingenieure, 1909)

Programmable robot, first century CE. Engineer: Heron of Alexandria

The first waves of automation from the second half of the nineteenth century meant many workers lost their jobs. Certainly from the moment robots started to be used in the production process. A robot is tuned and programmed so that it can execute a series of tasks repeatedly without getting tired or ill and without requiring holidays. In other words, robots carry out set algorithms.

The next step in the evolution of technology takes this several steps further. We are now building machines that can programme themselves. Moreover, smart algorithms are capable of teaching themselves tasks. Shazam, for instance, is able to recognise songs and continues to improve. Google Translate is gradually learning all the languages in the world, much like a child does.

Automating certain work or company processes is referred to as RPA (robotic process automation). This doesn't necessarily imply the use of robots but rather the replacement of human processing by software or other automated processes.

There are doomsday scenarios predicting that future software will be able to tackle any conceivable task and will quickly outpace human smarts.

Knowing, understanding

1. Briefly describe the key pivotal moments in automation history.
2. What important evolutions are on our doorstep in terms of automation?

Doing

3. Being able to give advice on automation in a given working environment.

Chapter 6

LEARNING MACHINES

- **Are computers really smart or do they only carry out programmed instructions?**
- **How smart is artificial intelligence?**
- **How do computers learn?**

The previous chapter shed light on what algorithms are and how they work. It also offered an introduction to a number of common algorithms.

Artificial intelligence. To many it sounds like a scary proposition. Will computers and robots dominate our lives? Is there any future left for us? Is there much point to studying if computers and robots will usurp our jobs? Will AI become smarter than us or have we already passed that point?

This module will tell you where we are in the development of artificial intelligence. How does AI actually work? Can artificial intelligence be compared to human intelligence? Do both forms of intelligence work in the same way?

Are there areas in which AI will never surpass us?

How do people and computers learn?

Computers carry out instruction sequences (**algorithms**) to turn input into the desired output. No matter how smart some of these algorithms, they cannot hold a candle to human intelligence. And of course algorithms were first conceived of by people. As of yet, computers don't manage to come up with all algorithms independently or offer the best possible solutions to problems.

People do. Even though people tend to make a considerable number of mistakes in life, in many cases lessons are learned from them. Of course, making mistakes is not a requirement for all learning. Most knowledge and skills we acquire by experience or example. And we all know repetition is useful in this process.

Artificial learning

Artificial intelligence refers to computer systems that learn how to arrive at solutions independently. We programme software to learn by experience or by example just as people and animals do. To this end, computer scientists use various **learning algorithms**. They try, for instance, to mimic in software the way neurons work in the human brain. Such a system is referred to as an **artificial neural network**.

Learning by example

Most things in life we learn by example. Even at school we must learn from our mistakes.

Suppose you have a pet cat and you keep telling your two-year-old son: "This is a cat." However, your son has never seen a dog. Chances are that in his first encounter with a dog he will say: "Look mum, cat!" To which you will reply: "That's not a cat, that's a dog."

By continuing to repeat this process, your son will soon learn the difference between a dog and a cat. He will gradually learn to distinguish the two and also understand that the neighbours' sheep are neither dogs nor cats. Children learn by 'supervised learning' (at school as well). They learn from their mistakes and from examples because somebody points these things out to them.

By programming neural networks into software, computers gain the ability to learn (machine learning). AI software is first fed **training sets** like for instance a series of dog pictures. By feeding the software masses of dog pictures, the artificial neural network is gradually able to distinguish dogs in other pictures.

Adding weight

We continuously have to make decisions throughout our lives. This process involves weighing a number of factors. What am I going to wear today? What gift shall I buy? What shall I eat this evening? Whom shall I vote for in the next elections? It involves various considerations and some of them have more weight than others in your final decision.

In the end, all the factors are added up. Some add more weight than others. A neural network (whether a bunch of neurons in our brain or an artificial neural network) is presented many inputs (considerations) which are each attributed a weight. The neural network then 'calculates' a solution which is emitted as output.

Careful: neural networks are not the only model by which to achieve machine learning (see chapter 8).

When you're deciding whether to go to work by bike or by car, you are weighing many factors: the weather, distance, cost, fatigue, whether or not you have to stop by the shop afterwards, etc. Some days you go by car, others you don't. Your final decision (output) depends on many inputs. The neurons in your brain go through many (statistical) considerations. Maybe your colleague goes by bike, while you go by car. He/she may have considered different factors. If you were soaked once by a downpour, you learn to check the weather forecast before getting on your bike.

1. Scan the QR code
2. Watch the videos
3. Carry out the accompanying assignment.

Supervised learning

It's not because an AI algorithm was fed an elaborate dataset/training set that it will work flawlessly from that moment on. An AI algorithm can learn from its mistakes. This stage still requires human intervention. People can still check whether the algorithm's decision was a correct decision. This is referred to as **controlled learning** or **supervised learning**.

A spam filter is a well-known example of AI software based on neural networks. Most emails containing the words 'casino' or 'penis enlargement' will be spam. But when a friend drops you a line saying "Up for an evening at the casino?", you don't want it landing in your spam folder. Or when your MP sends you an email about your reproductive organ that includes the phrase: "enlargement of the prostate", you don't want the combination of the words 'penis' and 'enlargement' to make this email fall victim to your spam folder. The artificial neural network must weigh many factors in its decision to mark it spam or not.

A spam filter is made by composing a list of suspicious words. Messages that contain one or more of these words may then be categorised as spam. In some emails suspicious words are classified as 'true negatives' (no spam), in others as 'true positives' (spam).

Other terms may lead to the classifications 'false positive' or 'false negative', the former referring to spam that ends up in your inbox and the latter to non-spam that ends up in your spam folder. In such cases, you can classify an email yourself and the spam filter learns it made a mistake which it will take into consideration in the future. This is how spam filters continue to improve and make fewer mistakes.

This is why you could say that AI software literally makes assessments based on what it has been taught. These assessments may turn out to be correct (true positive, true negative) but may also turn out to be incorrect (false positive or false negative). If the latter is true, adjustments must be made.

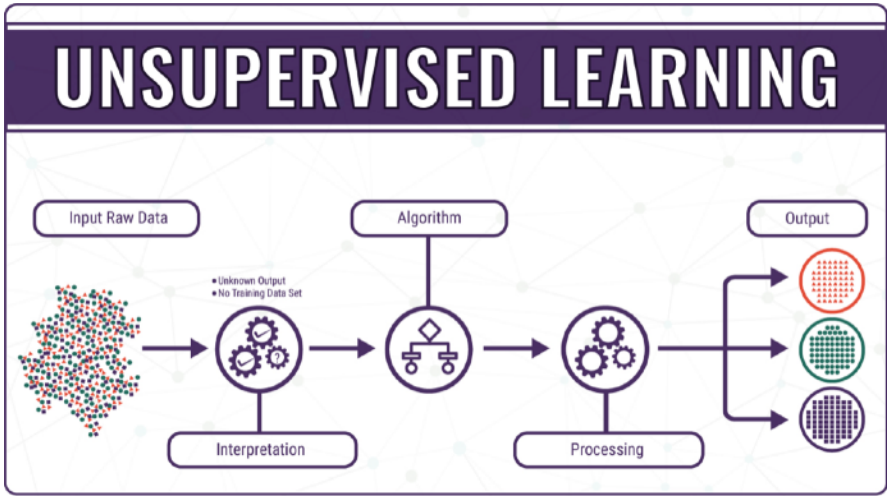
UNsupervised learning

In supervised learning, the algorithm learns from examples. In **reinforcement learning**, the algorithm is conditioned by confirmation or rejection. **Unsupervised learning** implies that the algorithm is not fed examples but instead must, for instance, divide elements into categories based on commonalities.

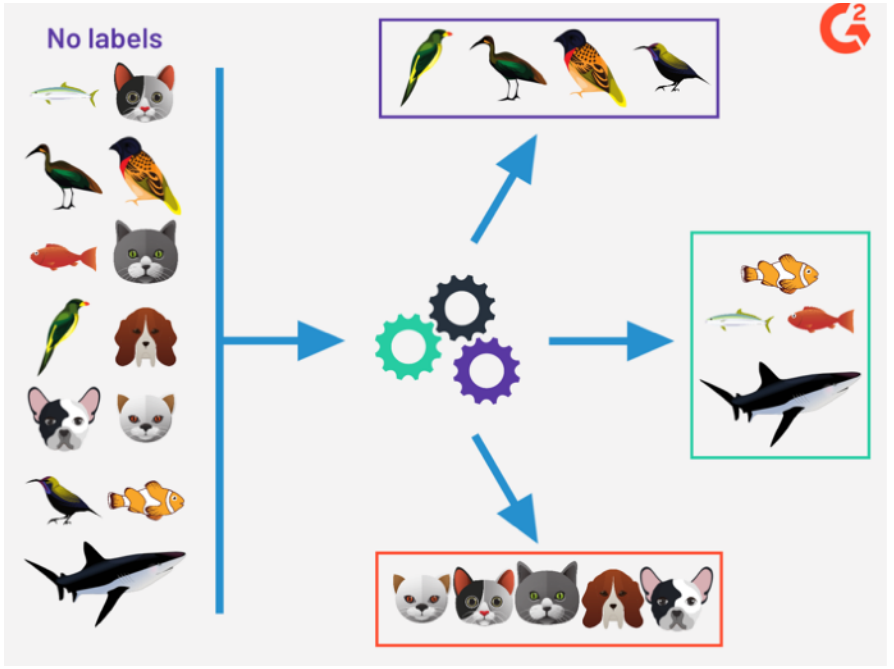
> Imagine having to do the laundry for the first time in your life. There is a mountain of laundry in front of you which must now be sorted based on a number of characteristics. The ideal output that it is sorted based on colour and textile. That is what the algorithm must now figure out. Based on the k-means algorithm, it can only be told beforehand you want to end with three groups. The rest it must figure out itself.

So with unsupervised learning, the algorithm is not given examples or labels beforehand. The algorithm must sort the information itself based on characteristics.

Chapter 8 (AI and data analysis) will tell you more about the various kinds of AI algorithms.



Source: https://medium.com/@chisoftware/supervised-vs-unsupervised-machine-learning-7f26118d5ee6



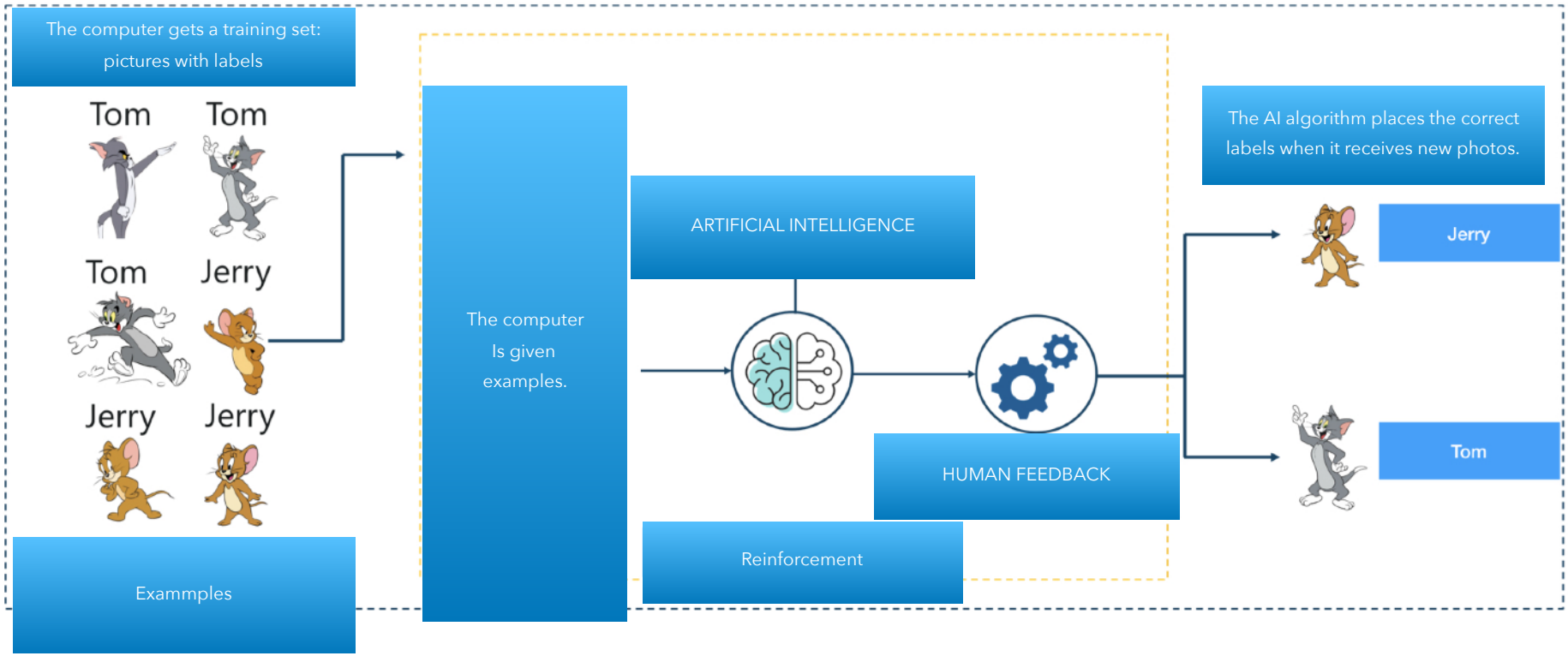
Source: https://learn.g2.com/supervised-vs-unsupervised-learning

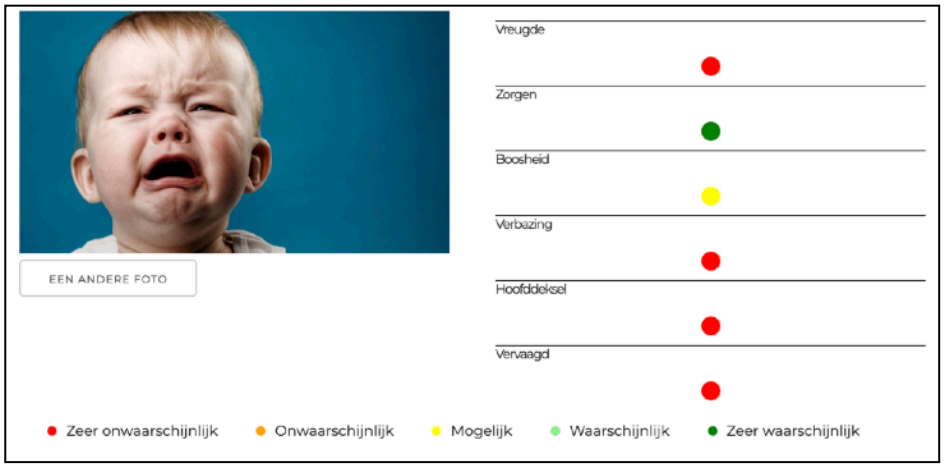
How can machines learn?

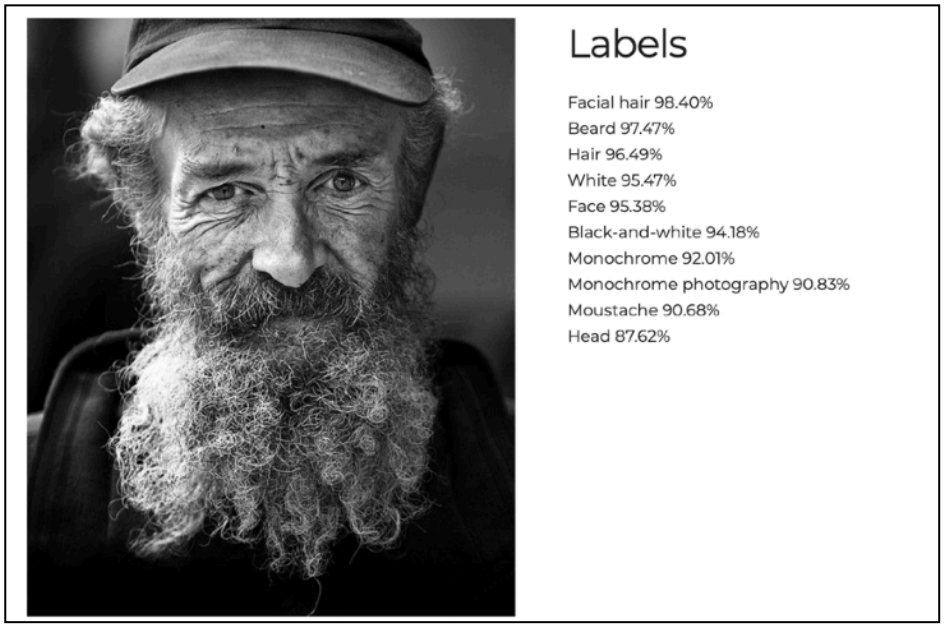
That is not something that is explained easily. Programmers often try to mimic a piece of the human brain. They try to code a brain cell (neuron). The resulting system is called an **artificial neural network**.

Sometimes artificial intelligence learns by **comparing**. When you watch a series on Netflix, Netflix compares your preferences and selections to those of other people who watched the same series. Netflix then suggests series that these other people watched as well.

Artificial intelligence learns by **probability**. For instance: there is a 50% chance that the first student you run into at school is a male and a 50% chance it is a female. However, in a convent of nuns it's almost 100% certain it will be a female. A computer wouldn't come up with that because it doesn't have that context available. When such information is fed into the AI software, it will come up with a more correct projection. You will learn more about the various kinds of AI algorithms in Chapter 8.







Google Cloud's Vision API is based on an AI algorithm that recognises emotions. It expresses these as statistical probabilities, in this case represented as colour values. (Source: www.schoolvoorbeeld.be/nodig/cloudvision - Kris Merckx)

An artificial neural network calculates statistical probabilities. (Source: www.schoolvoorbeeld.be/nodig/cloudvision - Kris Merckx)

AI is limited

Every bit of AI software and every artificial neural network must still be programmed by people. We still need to feed them training sets: **supervised learning**. A spam filter is good at recognising spam, but isn't able to drive a car. The software of such a car may be able to avoid hitting a deer crossing the road but it can't be set it a translation assignment. Every bit of AI software is still limited to **its own specific purpose**.

It is therefore safe to say that most AI is limited and that most AI algorithms can be categorised as **Narrow AI**.

AI set to conquer the world

And yet there seems to be no stopping it. Amazon, Netflix and Spotify offer their users **recommendations** based on the behaviour of other visitors. Self-driving cars anticipate the driving behaviour of cars in their vicinity to avoid accidents. Weather forecasts based on AI are more accurate then ever. Digital assistants like Google Home and Amazon Alexa are able to interpret (some) common questions and try to formulate a meaningful answer to them. Google Translate does a little bit better every day.

Artificial general intelligence

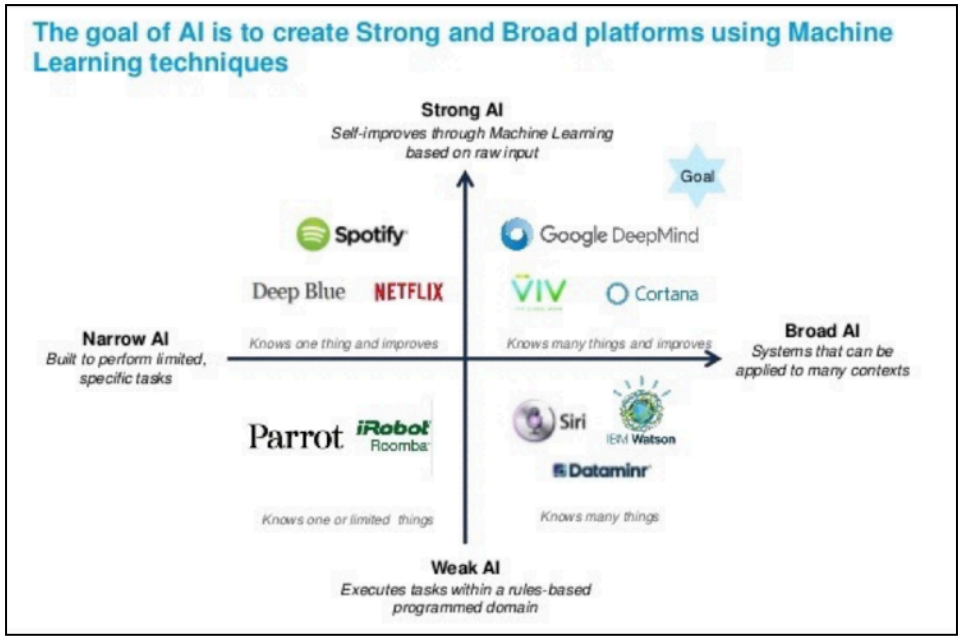
Those who fear that AI has overtaken us, can rest assured (for the time being). In spite of the inordinate enthusiasm and innumerable press releases, we are still a while away from an AI algorithm based on unsupervised learning that can tackle any problem (broad and strong AI or Artificial General Intelligence).

However, the development of unsupervised learning and deep learning is progressing. Algorithms learn how to solve problems autonomously without having to dig through training sets first. One of the rising stars behind deep learning is Demis Hassabis and his company Deepmind that was usurped by Google. Deepmind made headlines when the AI programme AlphaGo beat the number one player of the Chinese game Go.

Will people become redundant?

To give you an idea: in Go, a player can make a greater number of moves than there are atoms in the universe. AlphaGo's winning move took Go players around the world by surprise. No human player would have come up with such a brilliantly creative move.

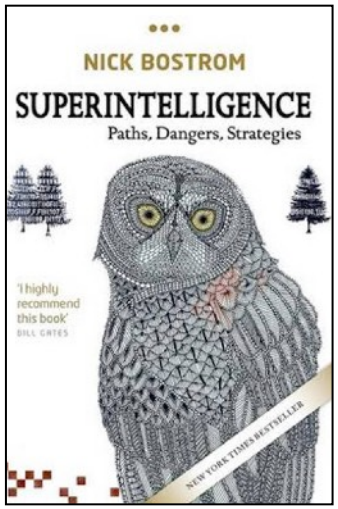
Automation made manual labour redundant in the past. AI can automate tasks which before required human brainwork. AI will certainly make many jobs and training programmes redundant. As to which jobs will be the first to bite the dust, there are conflicting reports. Generally speaking, it is believed that jobs



which require more expertise or creativity will survive longer as well as specialised manual labour..

(Source: AMP new ventures)

The book Superintelligence by British computer scientist Nick Bostrom



February 2018. According to the Economic Times of India, Dr David Hanson of Hanson Robotics claimed at the World Congress of Information Technology in Hyderabad that robots "would be alive and have full consciousness in five years".

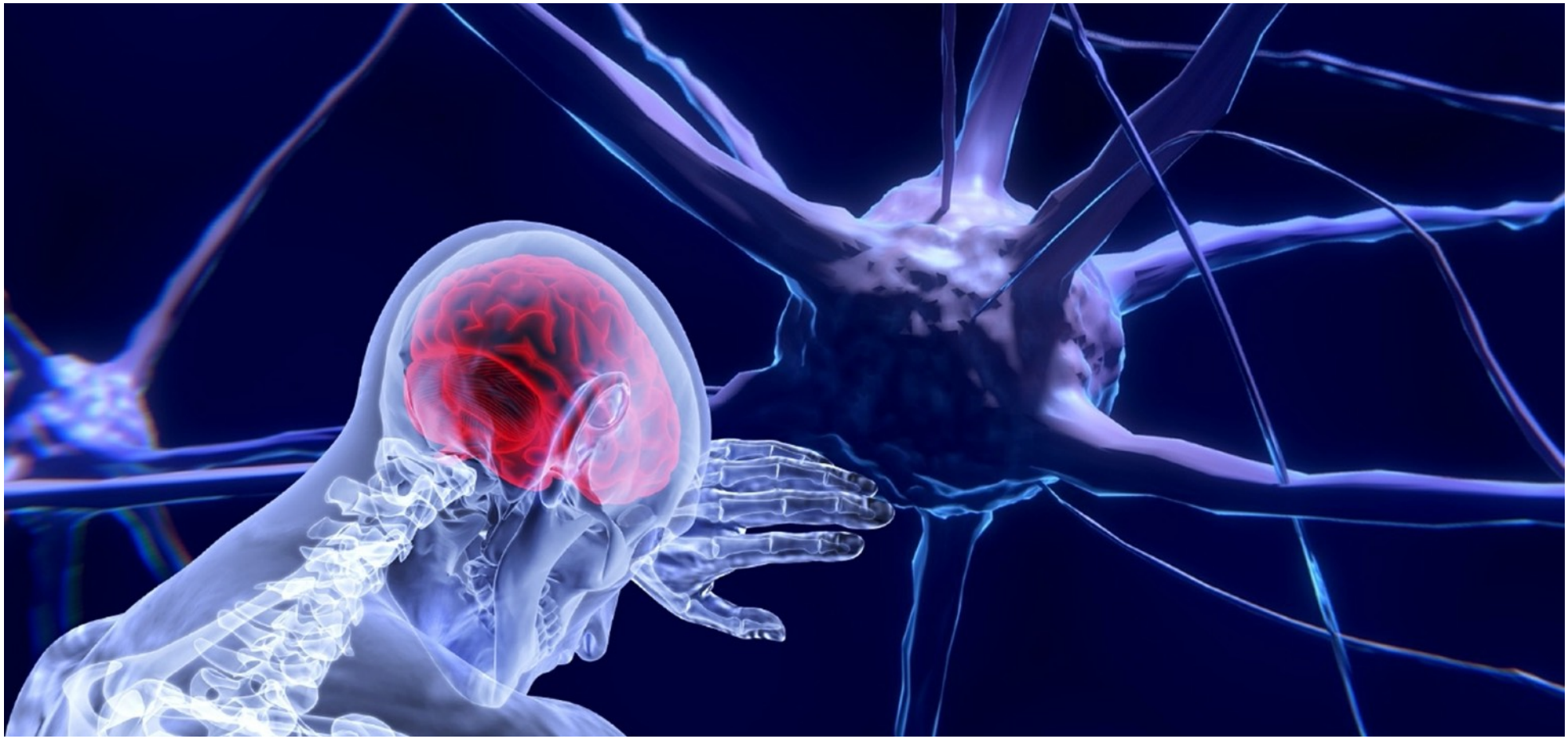
Singularity

When a computer achieves human intelligence, a level humans only reached after millions of years of biological evolution, it doesn't stop there for AI. Nick Bostrom, Ray Kurzweil, Bill Gates, Elon Musk and others have warned about this pivotal moment they refer to as singularity.

Artificial intelligence will, according to some, continue to develop at an exponential rate. They believe human intelligence is not a final destination but constitutes a pivotal moment from which point onwards human intelligence will quickly become redundant.

To Ray Kurzweill, singularity is the moment at which man and machine will morph together and hybrid transitional forms like humanoids or cyborgs will come into being.

Bostrom, Gates and Musk call upon governments to give this some serious thought and come up with rules and restrictions now. One of the first consequences of this call is an international willingness to put a stop to the development of autonomous robots used in warfare and on the battlefield.



Knowing, understanding

1. How can machines learn?
2. Explain the difference between supervised and unsupervised learning.
3. What is the use of training sets?
4. What is the purpose of artificial neural networks?
5. Explain briefly how artificial neural networks work.

Doing

6. Identify cases in which the use artificial neural networks make sense.
7. Come up with a structured approach to building a supervised neural network.

Chapter 7

AI AND THE NOTION OF IMAGE AND LANGUAGE

- **Why do computers and even AI have such difficulty with human language?**
- **Why is it difficult for AI to interpret what an image shows?**

Even though a computer can display any image imaginable, understanding what it depicts is entirely different.

Nor is it a walk in the park for computers to understand human language. To say this is simply down to the fact that computers only understand binary is a bit simplistic. That is only part of it. When a text is entered on a keyboard, the computer will translate every key stroke into a code. Every key on a keyboard is allocated a number internally. In other words, a computer doesn't understand what you are entering.

But the problem is much broader. There are about 6,500 human languages around the globe, each with their own vocabulary, spelling, grammar, etc. Every language has furthermore variations in the form of local dialects. Even though many languages are related to each other (e.g. the Indo-European languages), it is still far from obvious to make software understand all these languages. In addition to the western alphabet there are also other writing systems.

Furthermore, the written and spoken versions of languages vary and, quite aside from understanding language, generating language is also kind of tricky.

When a teacher asks you a question during an examination, the best case scenario is that you can answer it. You understand the question. You come up with the right analogy and correct parallels in your mind. You can rephrase it and put it in your own words. You don't learn the answer by heart and simply reproduce it without understanding what it means. Well... you could of course, but teachers expect you to draw intrinsic correlations and understand the meaning.

Some words have more than one meaning (e.g. 'grave' or 'lie'). Sometimes the meaning of a word can only be derived from its context.

> The dog sleeps in his basket. Jacky is very sweet. When I get home, he comes to meet me wagging his tail. What is keeping him today? Do you know? I'm so afraid he's done a bunk.

Read the short text above. As a human reader you instantly understand what this short text is about. How can a machine be taught to understand? From the get-go, there is an immense difference between man and machine. A child soon learns to associate the visual image of a dog to the notion of 'dog' and the sounds that go with it. Just think of parents going 'what is the sound a dog makes? No, not meow.' The child's language mechanism associates a dog with certain actions like barking, running, tail wagging as well as with body parts like tail, legs, tongue. It's an entire framework that fits in with the child's ontology of the world. But how can this way of processing language be taught to a computer? How indeed...

Let us first divide the text into its constituent parts: sentences (**sentence breaking**). This can simply be a matter of looking for punctuation marks like full stops, question marks and exclamation points. While this may do the job for most languages, it doesn't for languages like Arabic, Chinese or Japanese since they don't have punctuation. Subsequently, sentences could be split into the words they're made up of (**word segmentation**). Unfortunately, that doesn't work either for the languages above: Arabic, for instance, doesn't have space-separated words. Instead, it is the appearance of a given character that determines whether it's at the beginning, middle or end of a word.

Sentences may serve to share information or to request information (indicated by a question mark) and questions, in turn, can be open or closed (yes or no, if or else, 1 or 0). The process of identifying the nature of sentences is referred to as **discourse analysis**.

The machine will then turn to individual words and try to reduce these to their word stem in a process that is called **stemming** and **morphological segmentation**. In a language like English this is a relatively straightforward process:

open is the root of open, opens, opening, opened...

In a language like Turkish this is a very slow process since the stem of a word is not necessarily the same as its morphological root. It usually suffices that related words have the same stem. Many search engines treat words with the same stem as synonyms to arrive at a somewhat broader search (conflation). Google has been using stemming since 2003. Prior to that, the search term fish wouldn't have included fishing or fished.

Sometimes, this leads to erroneous results. The machine may offer too many results (overstemming) like when the search term university includes universe. In this case, the machine treats both words as synonyms because of their shared stem. At other times, there may be too few results (understemming).

Natural language processing involves identifying not only word stems but also word types (parts of speech): is it an adjective or a noun? Nouns furthermore include proper names.

***Jacky** is a dog.*

This technique of **named entity recognition** works by simply identifying capitals. But this, too, may be problematic. At the beginning of a sentence there is always a capital. In German all nouns begin with a capital letter.

***Jacky** ist ein **Hund**.*

And in English we use a capital when referring to ourselves.

***I**, **I** wanna be your dog.*

The part of speech is often stored in a kind of lexicon, a list established by identifying the part of speech based on the stem and any affixes (**part-of-speech tagging**).

This is not a straightforward process though. Indeed, depending on the sentence, a word may have entirely different meanings:

He lies in his grave because of his grave lies.

No doubt you'll have noticed, a word's meaning lies not only in itself but also in its relationship to other words like the **homonyms** in the example above. The process of distinguishing between such meanings is referred to as word sense disambiguation. The following sentence means different things entirely depending on whether you are commenting on a particularly lavish collection of typographical ornaments or of people behaving idiotically, something an NLP system has difficulty distinguishing between:

There are dingbats galore.

An NLP system uses a subset called natural language understanding (NLU) to try and find logic in short fragments of text. Which adjectives are associated with nouns (conference resolution)? How do the objects in a sentence relate (relationship extraction)? This is how an NLP system gradually tries to parse the sentence, create a family tree of a sentence and finally of the whole story.

> How do you like being a teacher? Oh, you know, there are dingbats galore but every now and then you come across an intelligent one. / There are dingbats galore for every possible use and occasion, the problem is picking one.

The excerpt/excerpts hints/hint at the sentiment behind it/them. An NLP system can make an educated guess of the sentiment that speaks from a text based on the meaning of words and the frequency by which they occur. This is referred to as **sentiment analysis**.

Spelling and grammar correction

Many word processors include spelling and grammar correction. MS Word for instance uses red wavy lines to indicate misspelled words. The programmers have incorporated the spelling rules into the software. Try to open a French or German text in a word processor set to English. It would appear everything is misspelled. That's because only the English spell check is loaded.

Thanks to the evolution of AI (artificial intelligence), **NLP (natural language processing)** has improved by leaps and bounds. Spelling and grammar rules are no longer coded per language (rule-based). The software learns about the rules from other texts. It also helps if the computer has a semantic understanding of the content.

Robot journalism

Even though understanding texts may still be relatively difficult for digital systems, an increasing number of articles are no longer written by people. Google acquired Jetpac, an app that automatically generates city guides based on image recognition algorithms. Associated Press has thousands of articles generated by robot writers.

Does that mean the end of the human journalist? The article below was written by a robot journalist:

"Aerie Pharmaceuticals Inc. (AERI) on Tuesday reported a loss of $13.1 million in its third quarter. The Research Triangle Park, North Carolina based company said it had a loss of 54 cents per share. Losses, adjusted for stock option expense, came to 44 cents per share."

The article meets the requirements for this kind of articles: it is correct and detailed. Of course a robot journalist is not a humanoid robot sitting in front of a keyboard. It's a piece of software that produces natural language. The author in this case is WordSmith from software company Automated Insights. In 2013, WordSmith generated 300 million news reports for various clients globally, more than all human journalists together and especially much cheaper.

According to Automated Insights, this doesn't jeopardise the jobs of real journalists:

"We're producing articles that never would have existed in the first place," he says. "AP was doing 300 corporate-earnings stories per quarter; they're now doing about 4,440. So 4,100 of these stories would not exist without Wordsmith." "The computer handles the who, what, where and when, and humans are freed up to ask why and how."

The LA Times uses Quakebot to deliver automated articles based on data of the US Geological Survey (earthquakes). Empathy, emotion or compassion are alien to this sort of article but that is not required for cut-and-dried factual reporting. For the moment, creative writing is beyond the realm of the possible because the algorithms are fed with standardised information, a bit like structured data in a spreadsheet. One possibility to add more human warmth to a text is by supplementing the data with sensor information.

It is often said that Wikipedia information is unreliable. Masses of articles, however, are no longer dispensed by human authors but by software algorithms. The best known Wikibot goes by the name of Lsjbot, programmed by the Swede Sverker Johansson. Lsjbot scrapes information from reliable sources and writes short articles about subjects related to animals. Every day, the bot creates about 10,000 articles. In total, about 8.5% of Swedish Wikipedia articles are written by Lsjbot.

Wikipedia used a bot for the first time in 2002. Going by the name of Rambot, it generated thousands of articles every day on just about every town, city or state in the United States as well as some other countries. A robot by the apt name of Asteroids scraped NASA data to write articles on... asteroids. Right now, there are over a thousand Wikibots at work: they improve texts and sort out inconsistencies. The most active amongst them is Cydebot which has made over 4.5 million modifications.

Still, the technique for the automatic generation of articles is not always indicative of artificial intelligence. For instance, Quakebot uses human-written templates and fills these with previously collected and structured data on earthquakes. The final result reads like an article written by people but, then, that is because a lot of it is. Essentially, the robot doesn't do much beyond filling templates with structured data. Other developers, like Narrative Science, try to raise the bar on robot journalism. They try to give software the ability to figure out what the data is about and generate content based on that. This means that the software also decides on the content it generates.

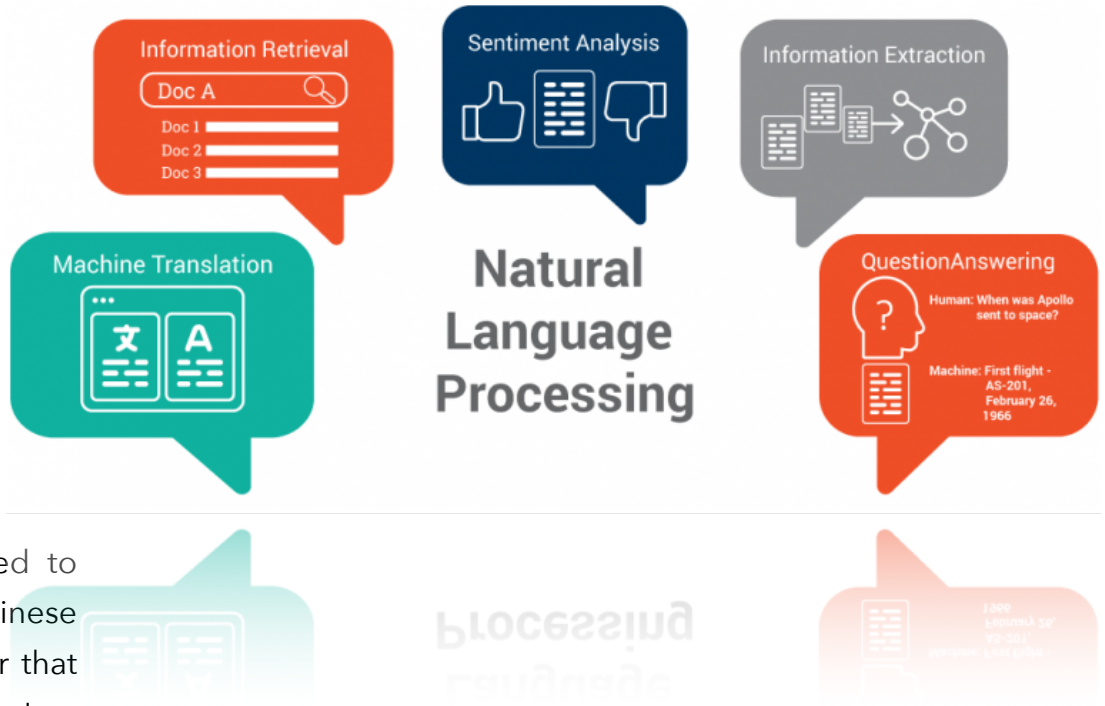
Spoken language and speech synthesis

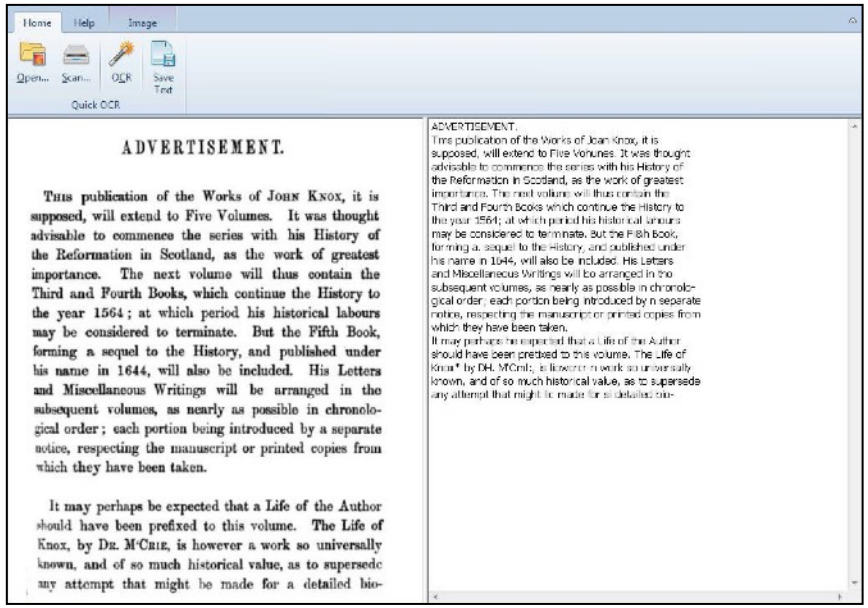
There is still a long way to go before machines are able to communicate flawlessly in human languages and this is not just about semantics. In the Studio 100 series ROX, the eponymous car communicates flawlessly with the other (human) characters. It recognises and interprets their questions and their answers and often reacts laconically. Our friend ROX manages to combine state-of-the-art NLP with speech recognition, speech segmentation and speech processing and tops it off with **natural language generation**.

Recognising spoken language (speech recognition) is considerably harder than reading or interpreting text because there are barely any pauses between words in spoken language. Also letters tend to blend together. Dividing speech into separate words (speech segmentation) is very difficult to programme.

Natural language generation on the other hand, has become very common in for instance GPS systems. The software can use previously recorded sound (separate words or sentences) and be combined into the desired phrasing. However, this is very different from generating speech artificially tone by tone (**speech synthesis**).

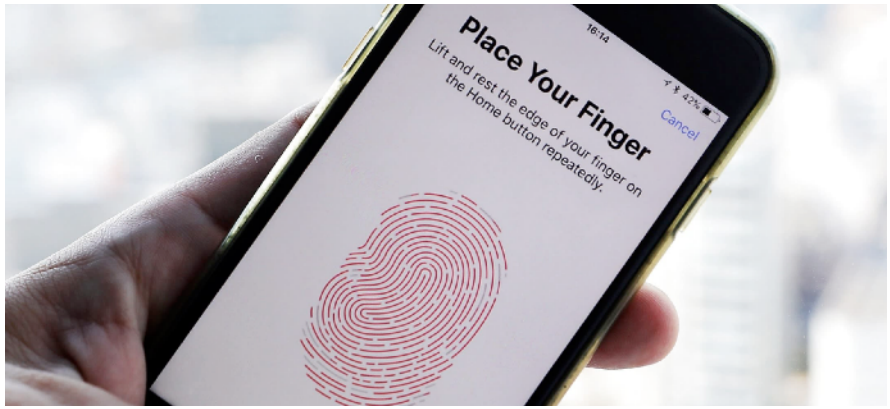
Machine translation joins together all the bells and whistles of NLP systems and natural language generation. Having a machine come up with meaningful translations is far from obvious. It requires not only an understanding of both languages but also a grasp of meaning and the ability to convert from one system of meaning (source language) to another (target language). For example, in Chinese there are no tenses so verbs don't give an indication of time frame. Tenses don't enter into it when writing or speaking Chinese, it is a completely different mind frame compared to speakers of Indo-European languages. Translating from Chinese to Dutch and vice versa is therefore far from easy. Consider that when people make fun of Google Translate. Google Translate furthermore works with machine learning. In other words, the software is continuously learning and keeps getting better.





Template-based OCR

(Source: https://www.malavida.com/en/soft/supergeek-free-document-ocr/#gref)



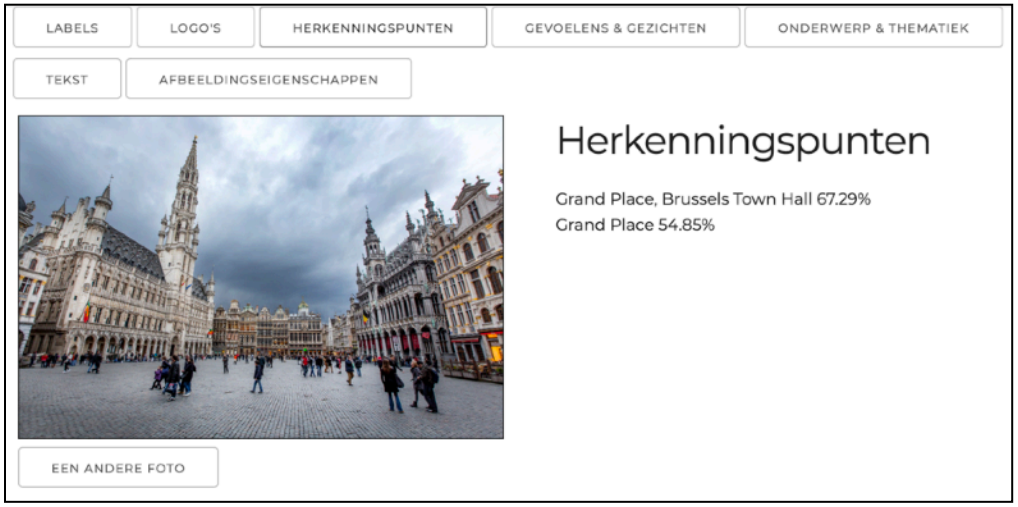
(Source: https://asia.nikkei.com)

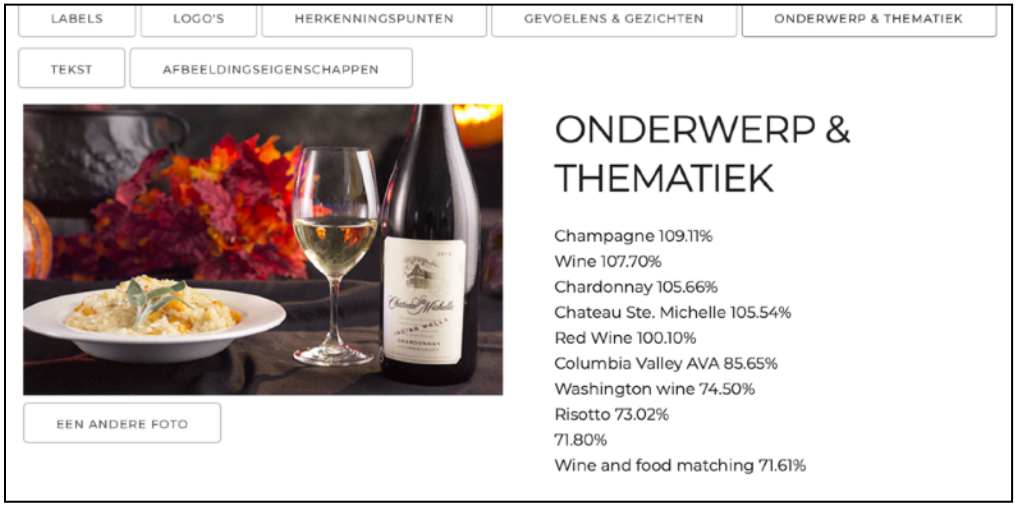
Image recognition

Your smartphone usually has at least one camera, many laptops are equipped with a webcam. This doesn't mean, however, that these devices can see like people or animals. We are not only able to see things, we can also recognise objects. Teaching computers to glean information from images is referred to as **computer vision** and **image recognition**.

In supermarkets there are scanners that recognise barcodes. Most smartphones can decipher QR codes. Smartphones can recognise the face or the fingerprint of users. Interpreting raw image data requires software. Just like people need more than eyes to recognise objects and people.

Barcodes, QR codes and text recognition are relatively old techniques now. The software reduces the image to a black and white image. **OCR (optical character recognition)** software tries to match groups of black pixels with existing characters allowing it to make a positive identification. That doesn't mean the software understands what the text is about.





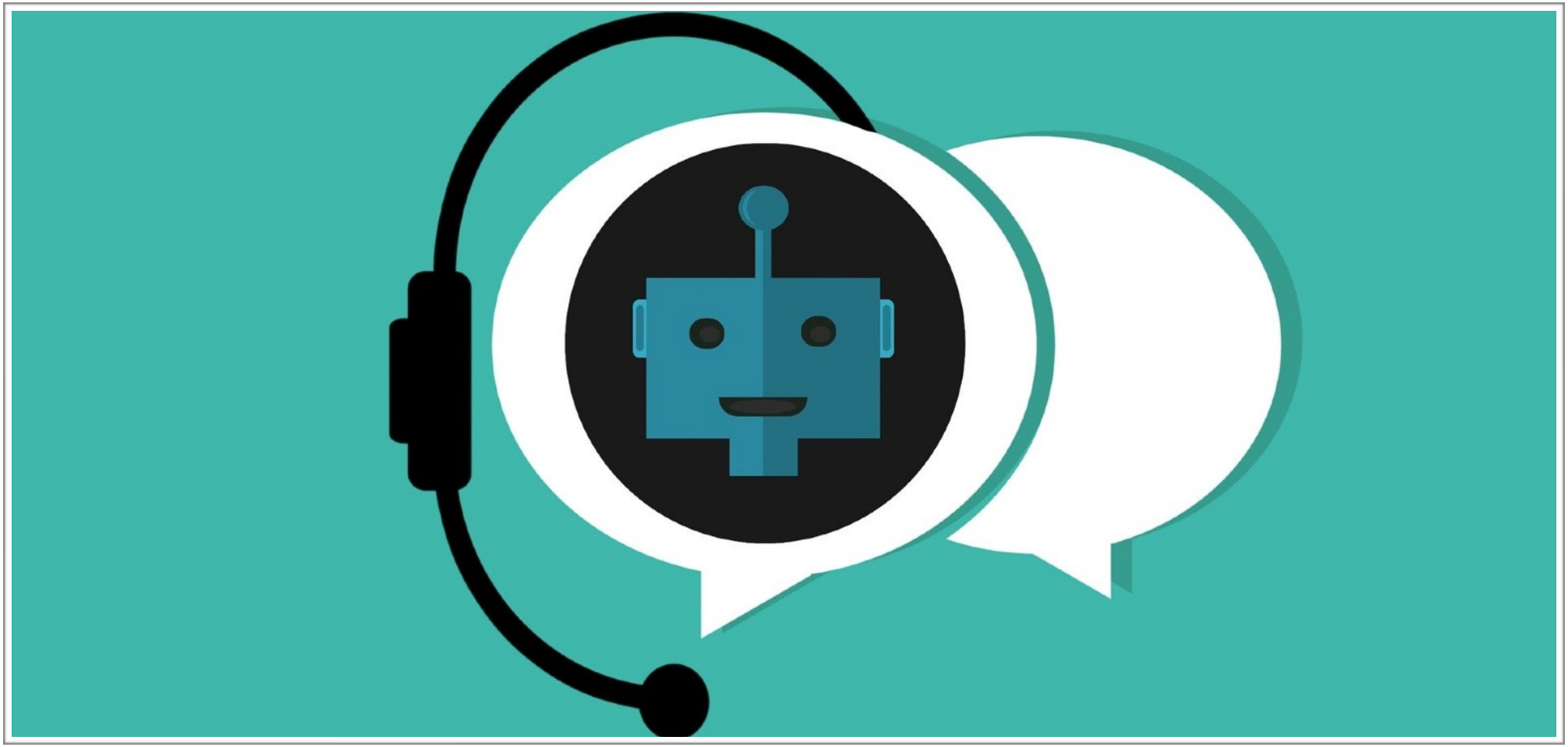
To teach computers to truly identify what is shown in an image, AI (artificial intelligence) is required. Just like a child needs to learn the names of objects and living creatures in the physical world, AI software must gradually learn how to recognise elements. When a child is shown certain letters frequently enough, it will start to recognise them. The same thing goes for AI. Classical OCR is unable to recognise letters that were printed upside down, but a child or AI is able to do so.

The section on AI will show you that AI is able to recognise images by mimicking the way neurons work in the brain. An **artificial neural network** learns to use labelled images to recognise images similar to these.

If you want to make a piece of software recognise a dog, it must first be fed hundreds or thousands of images with dogs in it. It allows software to learn gradually that both Pekinese and Pitbulls are dogs and furthermore that both Disney's Pluto and Lassie are dogs.

Only when a robot is able to recognise objects quickly and accurately will it be able to roam like real, flesh and blood people. Self-driving cars for instance, use AI-based computer vision and image recognition to recognise pedestrians, traffic signs, vehicles...

AI-based computer vision. The software uses the Google Vision API.
(Source: https://www.schoolvoorbeeld.be/nodig/cloudvision, Kris Merckx)



Knowing, understanding

1. Explain why computer systems have difficulty understanding human language.
2. Why do computers have difficulty tackling written language?
3. What is the difference between spelling and grammar correction on the one hand and NLP on the other?
4. At which point does a computer understand text?
5. Why is understanding text so hugely important in data analysis?
6. Why is it even more difficult to understand spoken text?
7. Why do computer systems have difficulty interpreting images?
8. Why is it fair to say that image recognition is a good example of artificial intelligence?

Chapter 8

AI AND DATA ANALYSIS

- What kind of algorithms are used in AI?
- What forms of analysis can AI be used for?

You already know that AI can learn from data in different ways, but what kind of algorithms or techniques (in addition to artificial neural networks) are behind supervised, unsupervised and reinforcement learning?

Without resorting to elaborate mathematical explanations, this chapter tries to give you a general overview.

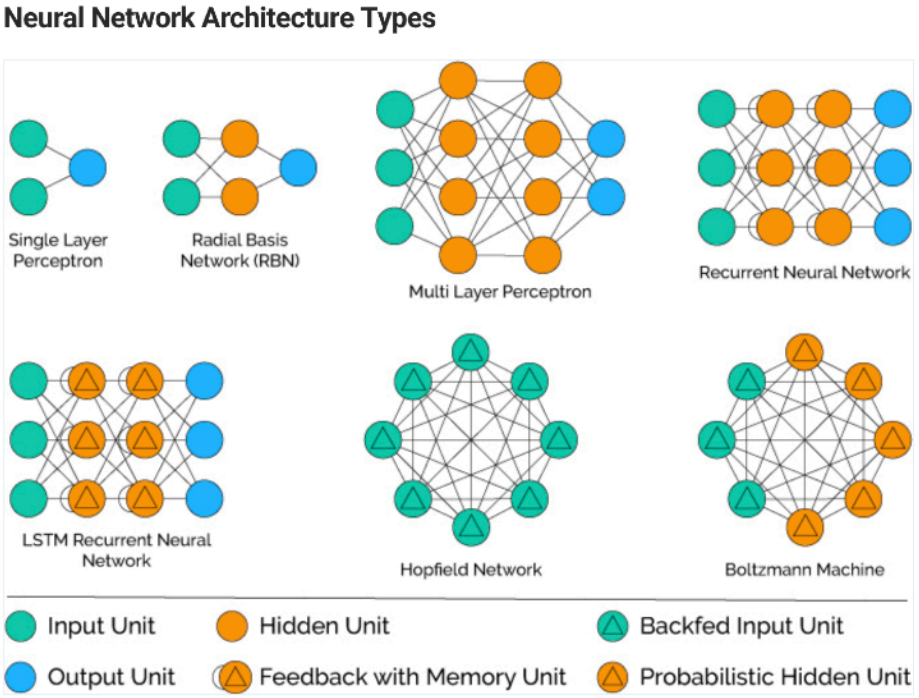
You will learn what ideas are behind heavyweights like k-means, linear regression, decision trees, random forest and support vector machines.

Neural networks

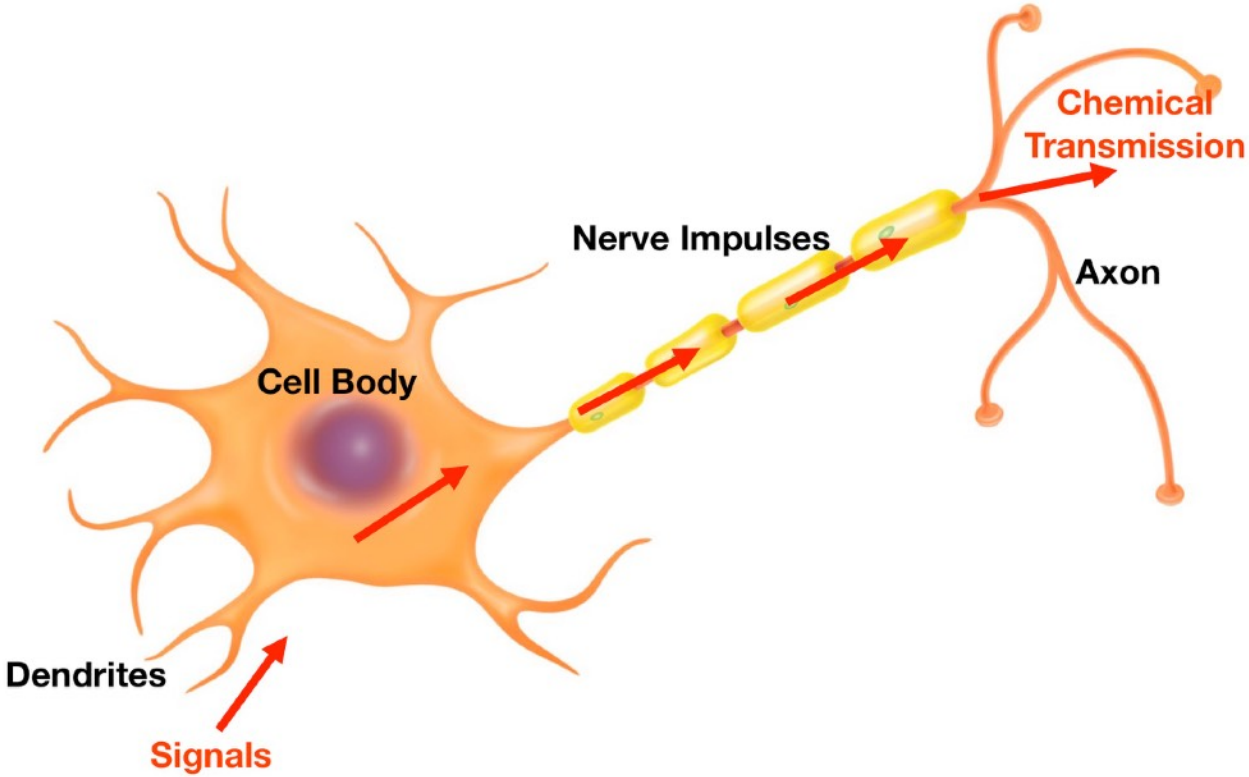
Artificial neural networks (ANNs) were already discussed in a previous chapter. ANNs are statistical models, inspired by and modelled after biological neural networks. Human brains contain an estimated 100 billion neurons. These neurons form networks that exchange information and form new ones based on that information. Precisely how all of this works is still not entirely clear, but learning could be a process of strengthening connections between neurons. Decisions (ranging from sticking it to someone to choosing between fries and a shawarma) are taken based on inputs received by the neurons. The output is the final decision made by the network.

ANNs are capable of establishing **non-linear correlations** between input data and output (the decision). Roughly, this means that the result is not always a choice in favour or against (so **not always a binary decision**) but rather a weighed assessment. The final decision is the result of adding up all the inputs according to the weight attributed to each of these inputs by the network (see the example of the bike and the car). ANNs use various algorithms and can take various shapes (architecture). To name a few: *Feedforward neural networks, Recurrent neural network, Multilayer perceptrons (MLP), Convolutional neural networks, Recursive neural networks, Deep belief networks, Convolutional deep belief networks, Self-organizing maps, Deep Boltzmann machines, Stacked denoising autoencoders, Backpropagation, Gradient descent algorithm, Hebbian learning rule, Self-organizing Kohonen rule, Hopfield network law, LMS algorithm (Least Mean Square), Competitive learning*, and so on.

Want to dive deeper into the rabbit hole of ANNs? Have a look at https://www.xenonstack.com/blog/artificial-neural-network-applications/



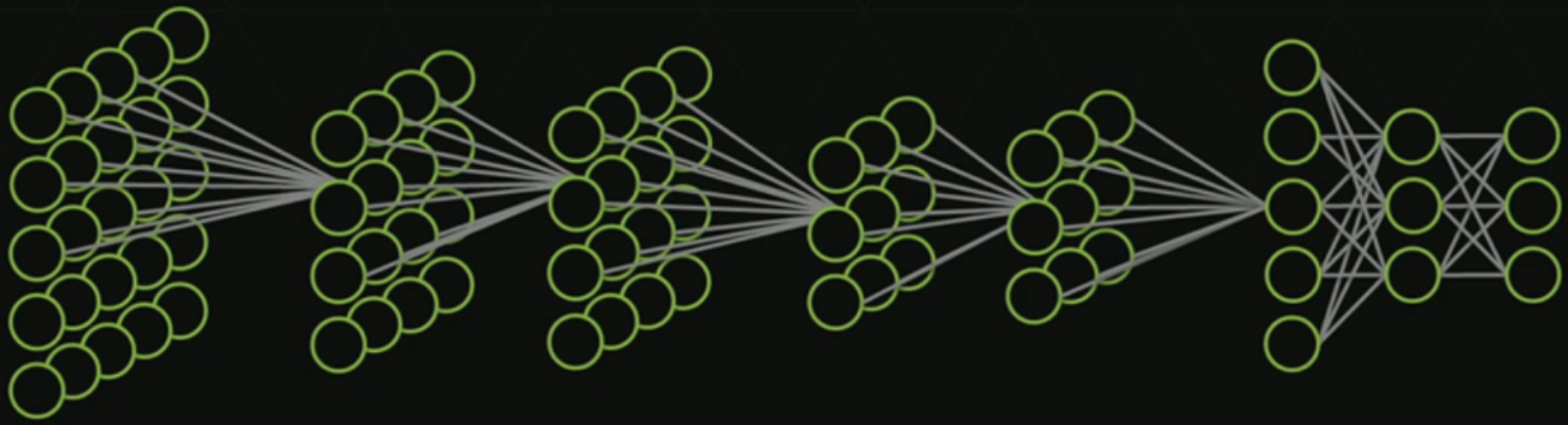
Source: https://www.xenonstack.com/blog/artificial-neural-network-applications/



A neuron in the human brain. Our brain contains an estimated 100 billion of these neurons. The neurons form networks and collect data together. (Source illustration: thinglink.com). A (multilayer) perceptron is an artificial neural network in which the neurons in one layer are connected to every neuron in the next layer. The first layer consists of input neurons where the input signals are established. This followed by one or more hidden layers which add greater learning ability and finally there is an output layer which renders the result produced by the perceptron. (Source: https://nl.wikipedia.org/wiki/Perceptron)

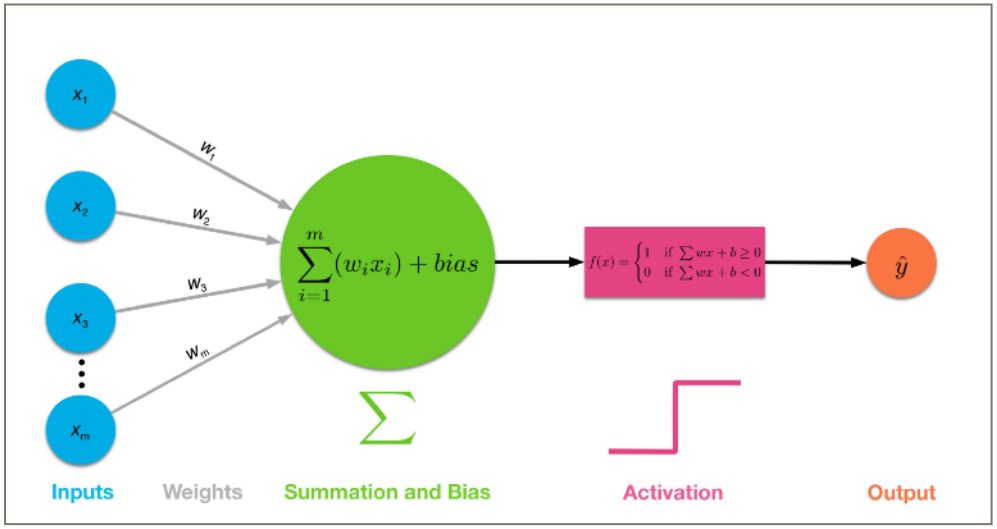
HOW A DEEP NEURAL NETWORK SEES

Image



"Audi A7"

Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.



A mathematical model of an ANN, in this case a single-layer perceptron.
Source: https://towardsdatascience.com/

The neural network in the illustration above contains a single layer, hence its name. This results in the following sigmoid function:

$(x_1 * w_1) + (w_2 * w_2) + \ldots + (x_n * w_n) + b = y$

The b=bias will be discussed in chapter 9.

Multilayer perceptrons and deep learning

A neural network comprising different layers is referred to as a **multilayer perceptron**. Look at the illustration on the previous page. Every layer of that network tries to recognise patterns in the image. When a pattern is found, the algorithm activates the next hidden layer. Say that the first layer recognises edges in the image. Then the following layer combines the edges found and so on. The more layers in the network (so the deeper the network), the better the pattern recognition. Every layer in the ANN attributes weights (or adds to it or subtracts from it), every time new data (in this case new images) enter the network which helps determine the input for the following layer.

Architecture	Example	Learning method
Convolutional Neural Networks	Image recognition	Supervised
Recurrent neural networks and long-short term memory	Natural language processing	Supervised
Autoencoders	detect abnormal data (for example fraud)	Unsupervised
Generative Adversarial Networks	Computer vision	Unsupervised

The combination of several multilayer perceptrons is referred to as deep learning. Deep learning takes different shapes and adopts different architectures.

AI as a solution to analysis problems

Essentially, all AI algorithms do the same. They predict an outcome (output) based on unknown input. However, it is the nature of the data that determines the most appropriate algorithm. AI algorithms can be classified based on purpose.

	Purpose	Learning method	Example
1	Classification	Supervised	Spam or no spam
2	Regression	Supervised	Predicting market value
3	Clustering	Unsupervised	Uncovering fraudulent transactions

1 Classification algorithms

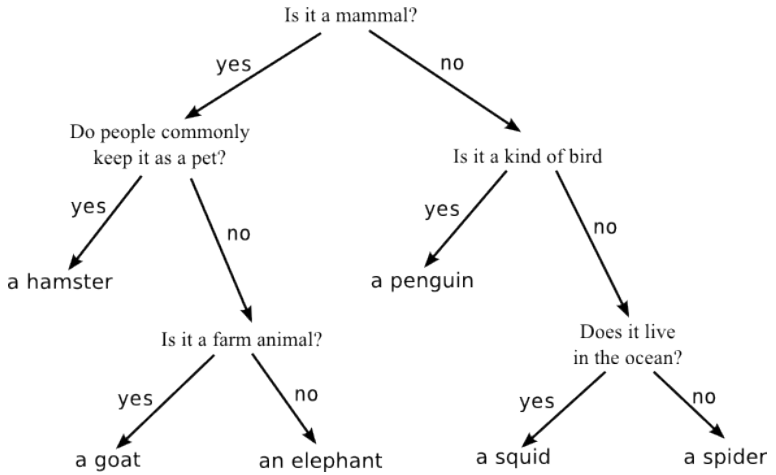
- Naive Bayes
- Decision Tree
- Random Forest
- Logistic Regression
- Support Vector Machines
- K Nearest Neighbours

Naive Bayes classifies data by assessing the statistical probability it belongs to a certain class or group. It does not consider correlations, it considers the number of commonalities and arrives at a probability rating based on that. The table below offers an example of sentiment analysis (on the left the dataset, on the right an interpretation)

TEXT	CATEGORY
I am so angry.	mood
I feel like superstar.	mood
It is going to rain.	not mood
I want to cry.	mood
They will come together.	not mood

I	mood
feel	mood
so	
terrible	

A **decision tree** has a structure akin to a family tree. Each branch tests whether a certain condition is met or an attribute possessed thus arriving at an output that determines the next branch. Careful, there is a risk of overfitting when additional input data may lead to radically different output (as with the training data here).



A **random forest** is a collection of decision trees in which each decision tree receives a subset of data attributes and provides an output based on this subset. The answer is then the mean of the outputs of all decision trees. This kind of algorithm reduces the overfitting problem and leads to a much more precise classification.

Simply put, **logistic regression** models the probability of what category data falls into based on other information. For instance: Predicting a student's likely final result based on previous results.

A **support vector machine** is a binary classifier, it divides entered data into two classes based on previously selected characteristics. An SVM must be trained with historical data before it is capable of this. The term 'vector' refers to the fact that each datapoint is given an x and a y coordinate in a graph (or vector space). SVMs work well with low-noise small datasets with barely any overlapping categories. They are not suitable for larger datasets.

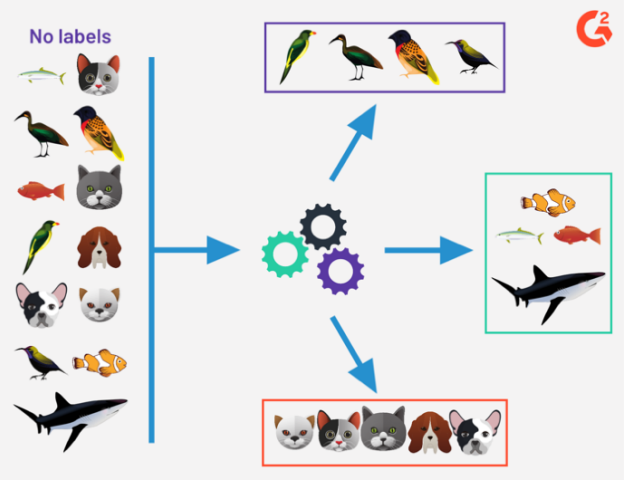
K-nearest neighbours (k-NN) starts from the assumption that the class an element belongs to corresponds to the classes of the elements nearest to it. Suppose that everyone in your environment votes green, there is a good chance you do as well. A k value of 5 means you look at the 5 nearest neighbours (in a vector space). When k has a value of 3, you look at the 3 nearest neighbours and so on.

2 Regression algorithms

Linear regression is the simplest and most effective of regression algorithms. It is used to weigh variables against one another (e.g. the cost of a house and the location of a house). This can be drawn on a graph to arrive at a regression line. House prices can be extrapolated from a linear regression based on location.

3 Clustering algorithms

The above-mentioned **k-means algorithm** divides data into a number of classes to be set by the user (k=1, k=4, etc.). In other words, the user determines the number of classes into which the data must be divided. It is a handy algorithm for datasets that has clearly distinguishable elements (points).



Knowing, understanding

1. Explain the difference between non-linear correlations and binary decisions using an example.
2. Give a clear explanation of the following concepts: ANNs, multilayer perceptrons, deep learning.
3. What three purposes can AI algorithms be used for?
4. Explain the learning method used by algorithms depending on the use.
5. List and explain six important classification algorithms.
6. Explain why in most cases clustering algorithms use unsupervised learning.
7. Name a practical application for regression algorithms.

Doing

8. Choose the most appropriate algorithm for a given dataset and problem definition. Make your case.
9. Choose the most appropriate AI learning method for a given dataset and problem definition. Make your case.

Chapter 9

BIAS IN ARTIFICIAL INTELLIGENCE

- **Are outcomes always objective?**
- **Does an algorithm always work flawlessly and reliably because it is based on math?**

In 2016, Microsoft unveiled Tay, a Twitter bot, as an experiment in conversational understanding. 'The more you chat with Tay', Microsoft claimed, 'the smarter it gets at involving people in an informal and playful conversation'.

Unfortunately, the conversations didn't remain playful for very long. Fairly soon after Tay was launched, people started to tweet the bot with all manner of misogynous, racist and Trump-like remarks. In under 24 hours, on a diet of extremist Twitter talk, Tay turned into a racist bully.

What went wrong? Clearly AI models don't always produce the hoped-for results.

The black box problem and XAI

Based on the training set and human feedback, the algorithm gave more weight to some elements and less to others. When the development team concludes that the control data produces the desired outcome, the model can be deployed in the real world. At that point, it is no longer the how and the why of the AI model's conclusions that are important. After all, the outcomes match the desired results. However, that is precisely where the problem lies. There is no arguing with AI models about the reason it came to this or that conclusion. The reason why seems inaccessible, something that is referred to as a **black box**. IBM tried to promote its Watson computer as an important tool in cancer detection, but the (human) oncologists simply didn't trust the system. Two oncologists can argue and discuss, not so with an AI model.

Increasingly, there is a demand that AI algorithms must also be explainable. The how and the why of a model's conclusion must be clear. Even AI critics and those who see a dystopian future ahead in which AI has taken over the world, are in favour of **Explainable AI (XAI).** Only by keeping AI algorithms and models explainable can the development of AI be kept under control.

In the Matrix trilogy, AI has taken over the world. People still believe they live in the real world but are in fact nothing more than avatars in a virtual reality.

Subjective results

Suppose that a credit company wants to know whether a person is creditworthy. A developer may use an AI algorithm to determine whether it is true for any given user. But creditworthiness is a subjective and vague concept. Someone is less creditworthy when the risk of late payments is greater (e.g. in the case of low wages or a high risk of unemployment). However, the provider of financial services can make a bigger buck out of this than out of traditional loans.

So as a matter of primary focus, the company will want to make as much profit as possible by granting as many loans as possible that must be repaid at the maximum interest. This way, 'creditworthiness' is redefined based on as high a profit margin as possible. If the algorithm finds that granting loans to less creditworthy people is an effective way to maximise profit, it will show predatory behaviour even if that wasn't the intention of the company.

Dataset bias

Bias may occur when the data that is collected is not entirely representative of the true situation or it reflects existing preconceptions. What if a piece of AI software is developed for recognising faces, but only feeds the system faces of white people as training data? Amazon developed a hiring tool that very clearly showed a bias for male candidates because it learned from the company's historical records.

Preparing training data

The attributes and characteristics used as training data will help determine the end result. Take for instance the Amazon hiring tool: you can use experience or training as input attributes. For creditworthiness, the determining factors are age and income and not so much profit targets.

AI bias: difficult to remedy

1 Unknown unknowns

Often developers only learn about the effects of the chosen attributes afterwards. In the case of Amazon, the developers reprogrammed the tool by keeping all gender-related words out of their data. But even then the modified system picked up on verbs that have a stronger correlation to men or women.

2 Training and control

Developers often divide their data beforehand into a training set and a control set. This means that the control data contains the same bias as the training data.

3 Lack of social context

Computer scientists or developers mainly want to develop systems that can be used in various situations. Precisely by leaving the social context out of the equation, bias is created. It is not possible to develop a hiring tool for Amazon and also use it for selecting and hiring construction workers

4 How to measure fairness?

How to define fairness and bias? Is the idea of Zwarte Piet racist? Bias is present always and everywhere, but everyone defines it differently. It is also present in AI data. The difference, however, is that it is defined mathematically in the case of AI. How do we prevent bias? Must all groups be equally represented in all situations? Must men and women, people with light or dark skin, etc. be represented equally in all situations? Must women score equally high or as frequently in recruitment procedures, even when other necessary qualities are not met?

AI researchers do their best to tackle the problem of AI bias. They try to develop algorithms that help detect and reduce hidden bias in training data or in the model, irrespective of data quality. There is no simple solution though, it's an ongoing process. When developing AI tools we must be aware of the fact that outcomes are not only dictated by mathematical algorithms. Just like with all other aspects of society, data analysis is never entirely objective.

Kris Merckx - september-november 2019

Knowing, understanding

1. Explain what the black box problem is in AI.
2. Why is the demand for explainable AI growing?
3. Give an example of a situation in which AI analysis may result in subjective outcomes.
4. In what ways may bias make it into datasets?
5. Give four reasons (and explain) why bias in AI is difficult to remedy.

Doing

6. Make an assessment of whether bias could be present in a given dataset.

www.schoolvoorbeeld.be

Image recognition tool
www.schoolvoorbeeld.be/nodig/cloudvision

Kris Merckx 2019

We hear about it repeatedly and frequently: artificial intelligence. What is it? How does it work? Why are we suddenly referring to computer systems as ‘intelligent’ or ‘smart’?

Should we be afraid of artificial intelligence? Will we, human beings, become superfluous? Will machines take our place?

Are there limitations to artificial intelligence?

Are there things artificial intelligence will never be able to do?

This book attempts to offer clear answers to a great many questions about artificial intelligence and data analysis.

Kris Merckx