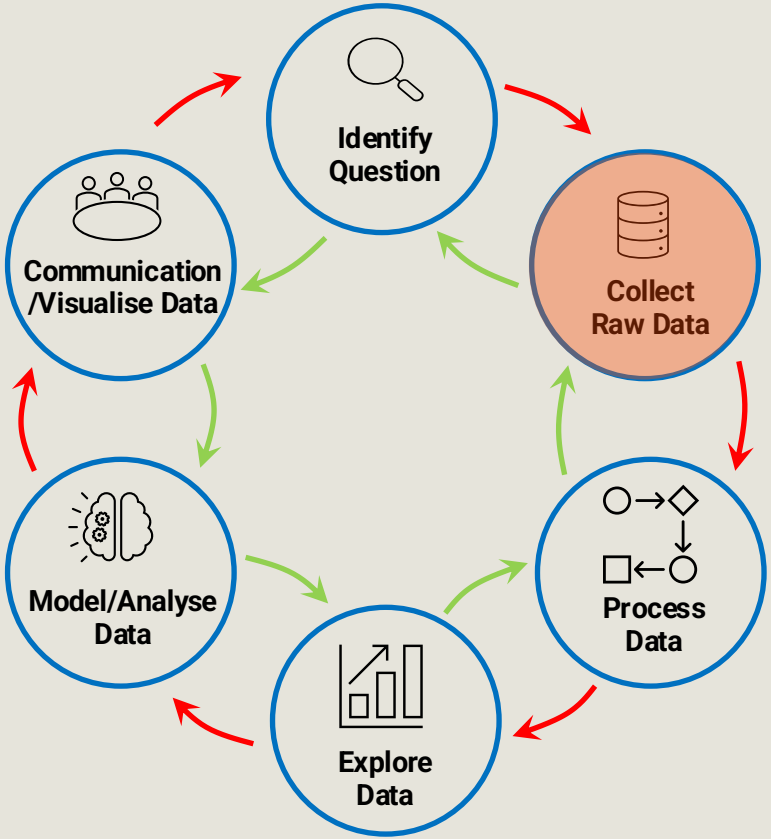
DATA – Introduction to Data Science

Semester 2 | 2025/26

Lecture 3 : Data Collection: Where Does Data Come From

Dr Xinwei Fang



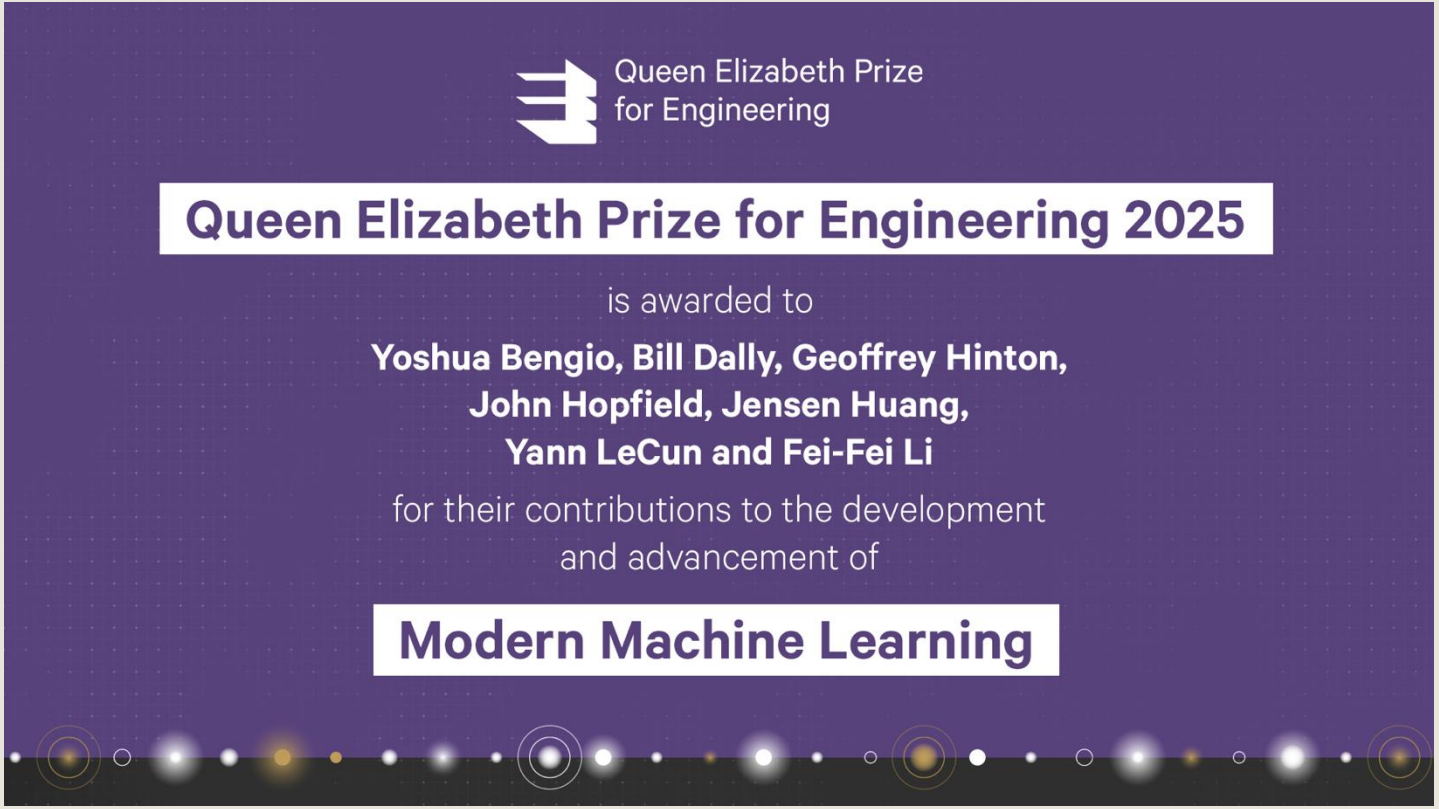
Collect Raw Data

- Decide how to collect the data
- Identify which data is relevant to the question
- Choose appropriate data sources such as surveys, databases, sensors, or logs
- Ensure data quality availability and ethical compliance

Lecture Overview

- Identify and distinguish between different types of data
- Recognise common sources where data can be collected
- Understand the key considerations involved in data collection

Queen Elizabeth Price for Engineering



Who are they?

Yoshua Bengio is one of the founders of deep learning. His work on neural networks, representation learning, and optimisation made it practical to train very deep models. Awarded the **2018 Turing Award** for contributions to deep learning.

Geoffrey Hinton A pioneer of neural networks whose ideas underpin modern deep learning, including backpropagation-based learning and energy-based models. Widely regarded as a key architect of today's AI systems. Awarded the **2018 Turing Award** and **2024 Nobel Prize**.

Yann LeCun A leading figure in machine learning and computer vision, best known for inventing convolutional neural networks. His work enabled practical image recognition at scale. Awarded the **2018 Turing Award**.

John Hopfield Introduced Hopfield networks, linking physics, optimisation, and neural computation. His theoretical models shaped modern thinking about memory, learning dynamics, and energy-based systems. Awarded the **2024 Nobel Prize**.

Bill Dally is an American computer scientist and educator, best known as Chief Scientist and Senior Vice President of **Research at NVIDIA**

Jensen Huang is the **founder and CEO of NVIDIA**, who transformed GPUs into the dominant platform for artificial intelligence.

Fei-Fei Li is a leading computer vision researcher who drove the data driven AI revolution through **ImageNet**.

"While a lot of people are paying attention to models, let's pay attention to data"

—Fei-Fei Li

ImageNet is an image database organised according to the WordNet hierarchy (currently **only the nouns**), in which each node of the hierarchy is depicted by hundreds and thousands of images.

Importance. The project has been instrumental in advancing computer vision and deep learning research. The data is available for free to researchers for non-commercial use.

The Scale of ImageNet

1.2 M+

Images with SIFT (Scale-Invariant Feature Transform) feature

14 M+

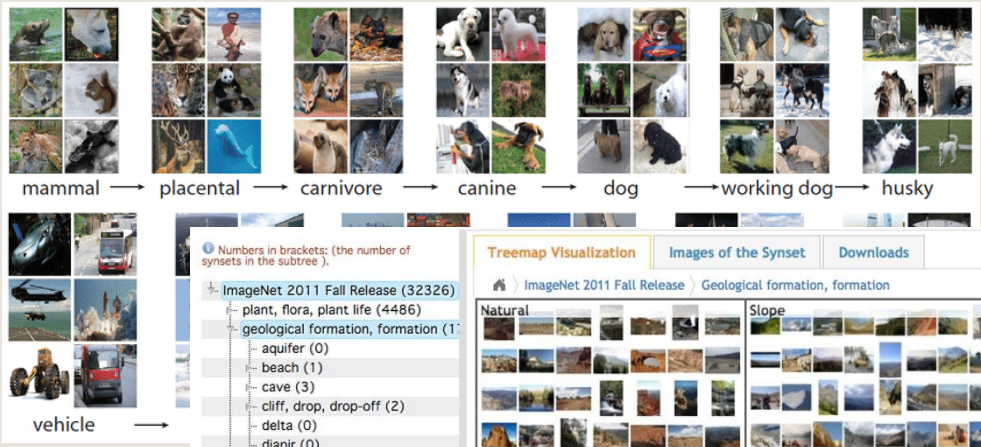
High-resolution images

1 M+

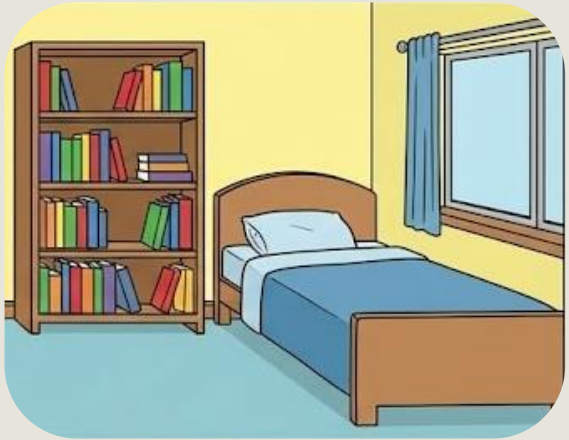
Images with bounding box annotations

20,000+

categories



Different data format (Unorganised Data)



Classifying the Raw Data

The Good, The Not-So-Bad, and the Ugly

Rigid, Database-ready

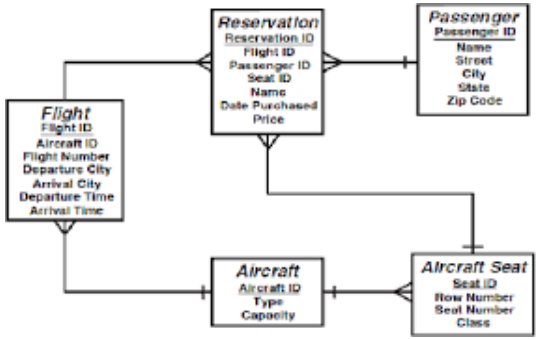
Flexible, Human-readable

Raw, Media heavy

The Good: Structured Data

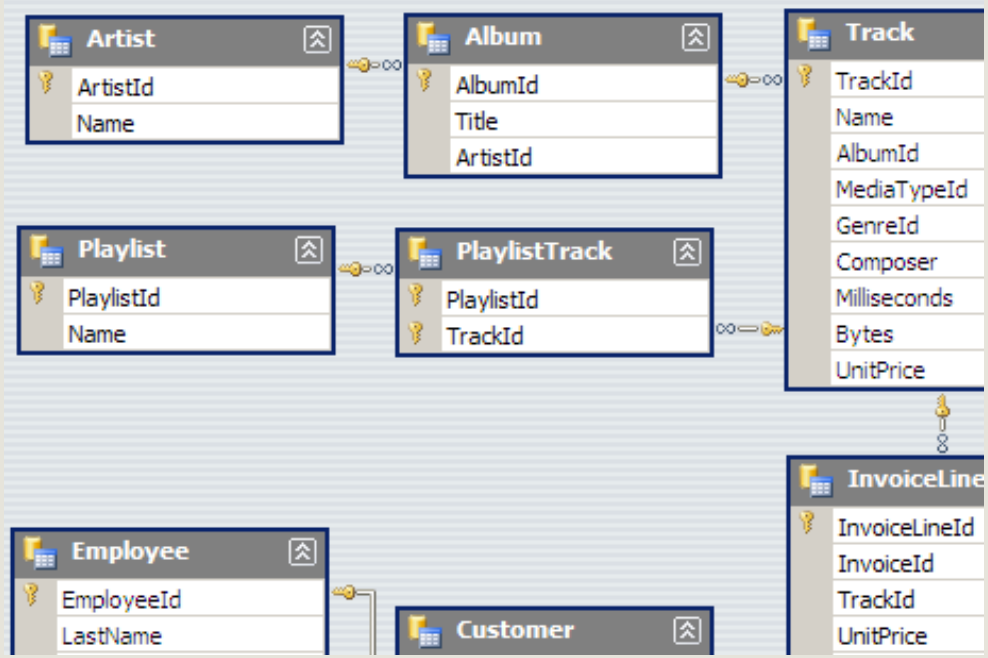
- Highly organised and neatly formatted.
- Conforms to a strict data model (numeric, date, address).
- Typically stored in relational databases

Benefits: Easy access, query, and analyse.



The Good: Structured Data

The data model representing a digital media store.



- Each table represents one real world concept only, such as **Artist**, **Album**, **Track**.
- Normalisation is applied - **No duplicated data stored in multiple places;** Each fact is stored once and referenced elsewhere; Updates only need to happen in one place
- Data types and attributes are meaningful - Each column has a clear purpose, such as **UnitPrice**, **Quantity**, or **InvoiceDate**. Attributes are atomic, meaning they store one value only, which improves querying and analysis.
- Scalable - New artists, albums, customers, or invoices can be added without changing the structure. This makes the schema suitable for long term use and real world systems.

Structured data is fundamental in large scale enterprise environments.

The Not-So-Bad: Semi-Structured Data

- Semi structured data **does not adhere to the rigid schema** enforced by traditional relational database models.
- It possesses certain **organisational characteristics** that make it suitable for querying and analysis.
- It uses **tags, keys, or other structural markers** to distinguish semantic elements and, in some cases, to define hierarchical relationships between records and fields.
- language independent (can be processed using C/C++, Python Java etc.)

Benefits: Flexible schema, Easier integration, Balance between structure and flexibility (e.g., CSV, XML, JSON documents).

The Not-So-Bad: Semi-Structured Data

	A	B	C	D	E	F	G
1	ID	country	points	price	province	tasterName	title
2	1	Portugal	87	15	Douro	Roger Voss	Quinta dos Avidagos 2011 Avidagos Red (Douro)
3	2	US	87	14	Oregon	Paul Gregutt	Rainstorm 2013 Pinot Gris (Willamette Valley)
4	3	US	87	13	Michigan	Alexander Peartree	St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)
5	4	US	87	65	Oregon	Paul Gregutt	Sweet Cheeks 2012 Vintner's Reserve Wild Child Block Pinot Noir (Willamette Valley)
6	5	Spain	87	15	Northern Spain	Michael Schachner	Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarra)
7	6	Italy	87	16	Sicily & Sardinia	Kerin O Keefe	Terre di Giurfo 2013 Belsito Frappato (Vittoria)
8	7	France	87	24	Alsace	Roger Voss	Trimbach 2012 Gewurztraminer (Alsace)
9	8	Germany	87	12	Rheinhessen	Anna Lee C. Iijima	Heinz Eifel 2013 Shine Gewurztraminer (Rheinhessen)
10	9	France	87	27	Alsace	Roger Voss	Jean-Baptiste Adam 2012 Les Natures Pinot Gris (Alsace)
11	10	US	87	19	California	Virginie Boone	Kirkland Signature 2011 Mountain Cuvee Cabernet Sauvignon (Napa Valley)
12	11	France	87	30	Alsace	Roger Voss	Leon Beyer 2012 Gewurztraminer (Alsace)

- **CSV: Comma Separated Values -** A plain text file with a list of data Each row (record) has fields separated by comma;

No enforced schema

	A	B	C	D
1	ID	country	points	price
2	1	Portugal	87	15
3	2	US	Eighty-Seven	14
4	3	US	eightyseven	13
5	4	US	87	65
6	5	Spain	eigthy-Seven	15

No relationships

Structured data in the strict database sense often means:
- Primary keys
- Foreign keys
- Constraints
- Relationships between tables

A CSV file does not store relationships explicitly.

No metadata

CSV does not store:
- Data types
- Constraints
- Null definitions
- Indexes

It is just plain text.

The Not-So-Bad: Semi-Structured Data

- JSON: Javascript Object Notation
- A lightweight data interchange format
- Stores data as key value pairs
- Human readable and logically structured
- Commonly used to transmit data between servers and web applications
- Language independent and supported by C, C++, Python, Java and many others

The Not-So-Bad: Semi-Structured Data

Why JSON is not structured?

```
[
  {
    "ID": 1,
    "country": "Portugal",
    "points": 87,
    "price": 15
  },
  {
    "ID": 2,
    "country": "US",
    "points": "eighty seven",
    "price": 14
  }
]
```

One ‘points’ is an integer, another one is a string

```
[
  {
    "ID": 1,
    "country": "Portugal",
    "points": 87,
    "price": 15
  },
  {
    "ID": 2,
    "country": "US"
  }
]
```

Second object has less fields

```
[
  {
    "ID": 1,
    "country": "Portugal",
    "points": 87
  },
  {
    "identifier": 2,
    "country": "US",
    "points": 87
  }
]
```

‘ID’ becomes ‘identifier’

JSON enforces:

- **Syntax rules**
- **Data types at value level** (String; Number; Boolean; Null; Object; and Array)
- **Bracket matching**

JSON does **NOT** enforce:

- Same attributes per object
- Same data types per attribute
- Mandatory fields
- Fixed schema

The Not-So-Bad: Semi-Structured Data

- XML: Extensible Markup Language
- Designed to store and transfer data
- Uses custom defined tags to structure information
- Extensible, meaning users can define their own tags
- Both human readable and machine readable
- Language independent and supported by C, C++, Python, Java and many others

```xml
<?xml version="1.0" encoding="windows-1252" standalone="yes"?>
<Records>
  <Record>
    <Row
      A="ID"
      B="country"
      C="points"
      D="price"
      E="province"
      F="tasterName"
      G="title"
      H="variety"
      I="winery"
    />
  </Record>
  <Record>
    <Row
      A="1"
      B="Portugal"
      C="87"
      D="15"
      E="Douro"
      F="Roger Voss"
      G="Quinta dos Avidagos 2011 Avidagos Red (Douro)"
      H="Portuguese Red"
      I="Quinta dos Avidagos"
    />
  </Record>
```

The Not-So-Bad: Semi-Structured Data

Why do we need both XML and JSON?

```xml
<review>
  This wine scored <points>87</points> in the competition.
</review>
```

XML allows structured data inside text.

```json
{
  "review": "This wine scored 87 in the competition."
}
```

JSON loses inline structure

```json
{
  "countries": ["Portugal", "US", "France"]
}
```

Array is explicit and compact.

```xml
<countries>
  <country>Portugal</country>
  <country>US</country>
  <country>France</country>
</countries>
```

XML must repeat elements manually.

CSV vs JSON vs XML

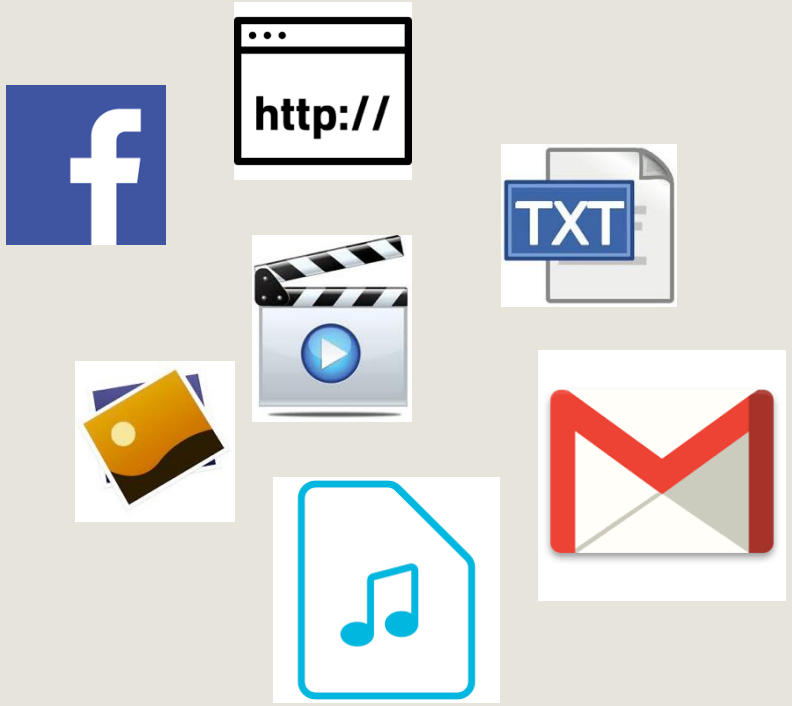
Feature	CSV	JSON	XML
Works well for spreadsheet style data	Yes	Yes	Yes
Can represent nested data	NO	Yes	Yes
Can mix structured data inside sentences	NO	NO	Yes
Can separate metadata from main data	NO	NO	Yes
Has built in data types such as number and true false	NO	Yes	NO
Built in name conflict prevention	NO	NO	Yes

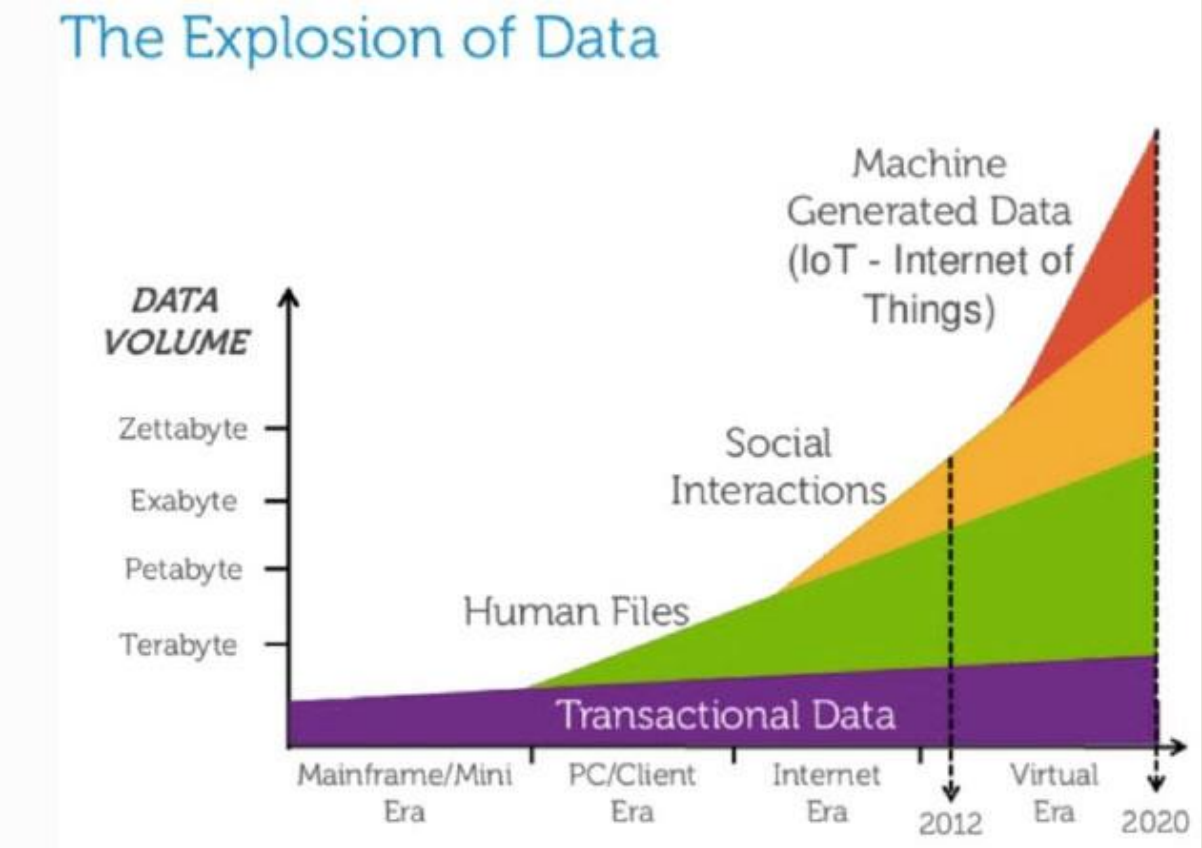
Conversions

	Conditions and Risks
CSV ⟶ JSON	**Possible if:** data is tabular, each row becomes an *object*. Information will be preserved.
JSON ⟶ CSV	**Possible only if**: flat structure, no nesting, consistent fields. Nested objects will be lost or flattened.
JSON ⟶ XML	**Usually straightforward.** *Objects* becomes *elements*, *arrays* becomes *repeated elements*. Structure mostly preserved.
XML ⟶ JSON	**Possible, but**: Attributes may be merged; Mixed content may be lost; Prevention of name conflicts may be ignored
XML ⟶ CSV	**Possible, but:** No hierarchy; No attributes; No nesting. XML becomes flat.
CSV ⟶ XML	**Possible if:** data is tabular, each row becomes an *element*. Information will be preserved.

The Ugly: Unstructured Data

- May have its own internal structure, but does not fit neatly into a spreadsheet or database.
- There is no standardised way to store this type of data.
- It requires additional processing before it can be analysed.





We are witnessing an explosion of data, but **80% of it is unstructured**. It does not fit neatly into a spreadsheet or database rows.

This data includes social interactions (tweets, posts), machine-generated logs (IoT), media (audio/video), and human files (emails, PDFs).

The Difference

STRUCTURED & SEMI-STRUCTURED DATA

Nicely formatted

Examples: Housing sell record, Product inventories

	Suburb	Addrsss	Roems	Type	Price	Method	SellerG	Dsts	Postcods	Regiennams	Propertycount	Bistance	CouncilArsa
0	Abbetsford	49 Lithgow St	3	h	1480000.0	S	Jells	1/04/2017	3067	Northern Mcnopolitan	4019	3.0	Yena City Council
1	Abbetsford	58A Turner St	3	h	1222000.0	S	Marshall	1/04/2017	3067	Northern Mstiopolitan	4019	3.0	Yena City Council
2	Abbetsford	1198 Yana St	3	h	1420000.0	S	Nelsen	1/84/2017	3067	Northern Mchopolitan	4019	3.0	Yana City Council
3	Aberfieldie	68 Vids St	3	h	1515000.0	S	Barry	1/04/2017	3940	Western Mchopolitsn	1543	7.5	Moones Valley Oty Council
4	Airport West	92 Clydesdale Rd	2	h	670000.0	S	Nelson	1/04/2017	3042	Western Mebopolitan	3464	10.4	Moones Valley City Council

UNSTRUCTURED DATA

Textual or non-textual, human or machine-generated. Requires heavy processing to be useful.

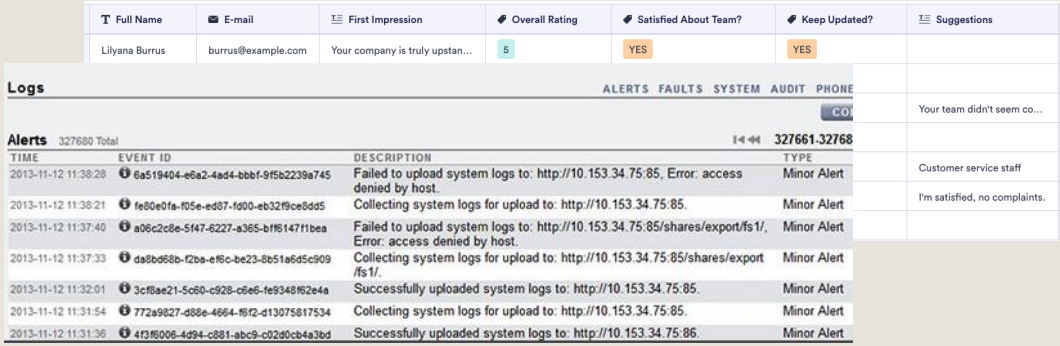
Examples: Chat logs, call recordings, social media posts, Images.

Structured data is **easier to analyse**, while unstructured data contains **more hidden value** but requires **more effort to analysis**.

Deciding where to collect

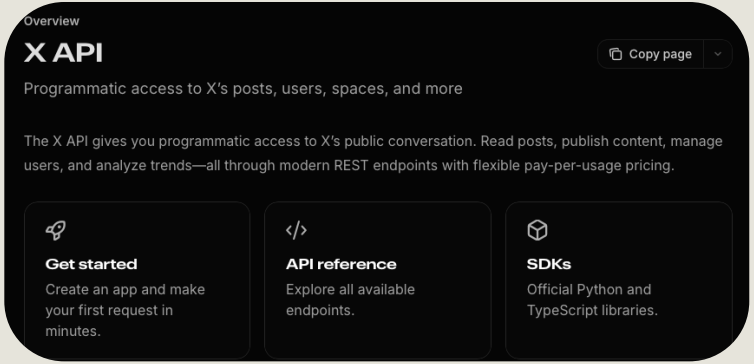
Internal Sources

Data already collected by the business.
Low effort, high access.
Examples: Customer service logs, transaction databases, operational records.



External Sources

Data held by outside entities.
High effort, low access.
Potential sources: API, Sensor deployment, Survey.



Internal data is usually **easier and faster** to work with, while external data can **add value** but often requires **more time and effort** to obtain and use effectively.

Existing External Sources Example - Application Programming Interface (API)

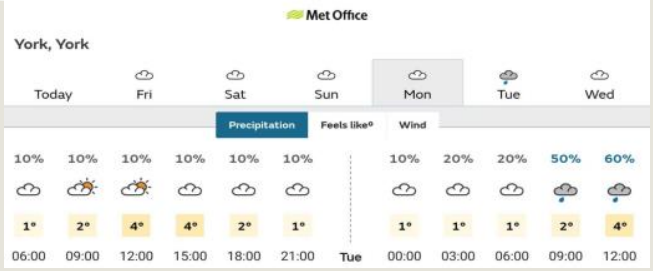
- An API allows one software system to request data or services from another system in a structured and controlled way.
- In data science, APIs are commonly used to:
  - Collect real time data
  - Access structured datasets
  - Retrieve metadata
  - Automate data pipelines

Using Met Office's API

THE SURFACE (What Users See)

Website

Mobile APP



Visual Weather Forecast: An intuitive, user-friendly interface presenting processed meteorological data.

Behind the Scenes (What Machines Read)

```
-<SiteRep>
  -<Wx>
     <Param name="F" units="C">Feels Like Temperature</Param>
     <Param name="G" units="mph">Wind Gust</Param>
     <Param name="H" units="%">Screen Relative Humidity</Param>
     <Param name="T" units="C">Temperature</Param>
     <Param name="V" units="">Visibility</Param>
     <Param name="D" units="compass">Wind Direction</Param>
     <Param name="S" units="mph">Wind Speed</Param>
     <Param name="U" units="">Max UV Index</Param>
     <Param name="W" units="">Weather Type</Param>
     <Param name="Pp" units="%">Precipitation Probability</Param>
  </Wx>
  -<DV dataDate="2021-01-28T14:00:00Z" type="Forecast">
    -<Location i="310169" lat="53.9621" lon="-1.0789" name="YORK" country="ENGLAND" continent="E
      -<Period type="Day" value="2021-01-28Z">
         <Rep D="ENE" F="-1" G="18" H="96" Pp="89" S="9" T="3" V="GO" W="15" U="1">540</Rep>
         <Rep D="E" F="2" G="13" H="99" Pp="42" S="4" T="4" V="MO" W="8" U="1">720</Rep>
         <Rep D="SE" F="2" G="11" H="99" Pp="25" S="4" T="4" V="MO" W="8" U="1">900</Rep>
         <Rep D="SSE" F="3" G="16" H="98" Pp="24" S="7" T="5" V="PO" W="5" U="0">1080</Rep>
         <Rep D="SE" F="4" G="18" H="98" Pp="95" S="4" T="6" V="PO" W="15" U="0">1260</Rep>
      </Period>
```

Raw XML Data: The underlying structured information, including specific tags that drives the visual output.

API Challenges and Considerations

Fundamentals

X API Rate Limits

Per-endpoint rate limits for X API v2

Rate limits control the number of requests you can make to each endpoint. Exceeding limits results in a 429 error until the window resets.

Limited API rate

DataPoint September 2025 Retirement FAQs

As of September 2025, Met Office DataPoint will be retired. Please see below the most recent frequently asked questions (FAQs) about this change.

Change of service without notice

Credit consumption details

Transparent pricing below. Pay only for what you use.

Resource	Unit Cost	Estimated Cost (per month)
Posts: Read Charged per resource fetched.	$0.005 per resource	Usage 10k resources 0k $50.00 50k
User: Read Charged per resource fetched.	$0.010 per resource	Usage 5k resources 0k $50.00 50k
DM Event: Read Charged per resource fetched.	$0.010 per resource	Usage 2k resources 0k $20.00 50k
Content: Create Creating posts or media. Charged per request.	$0.010 per request	Usage 5k requests 0k $50.00 50k
DM Interaction: Create Creating DM interactions. Charged per request.	$0.015 per request	Usage 1k requests 0k $15.00 50k
User Interaction: Create Creating user interactions. Charged per request.	$0.015 per request	Usage 2k requests 0k $30.00 50k

Exploding costs

Existing External Sources Example - Web Scraping

When legacy websites, smaller news sites, or government portals **do not offer APIs** (or RSS feeds), or when **API costs are prohibitive**, we use scraping.

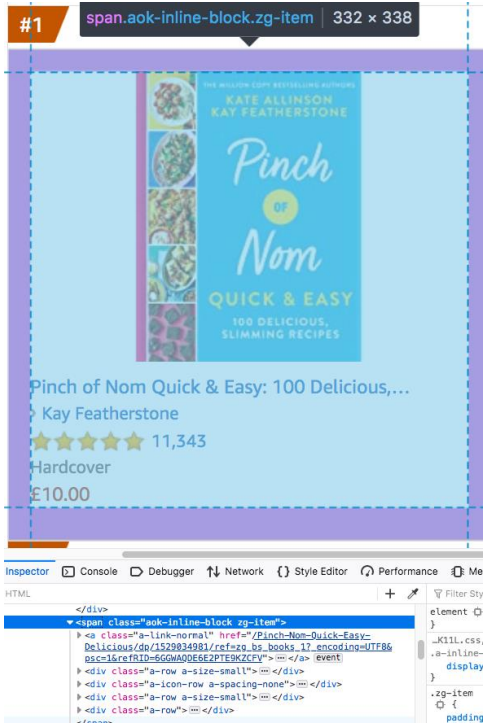
Use software (scripts) to extract data from a webpage

Specialised knowledge is needed to manually understand and identify that necessary data from a website

The data is extracted and saved locally for further processing

Example: Amazon

1. Find the HTML tags that hold the required data



2 . Parse the website using suitable Python libraries (e.g., BeautifulSoup)

```
def getAmazonBestSellers(pageNo):
    headers = {"User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gec

    urlTemplate = 'https://www.amazon.co.uk/Best-Sellers-Books/zgbs/books/ref=zg_bs

    r = requests.get(urlTemplate)

    content = r.content

    soup = BeautifulSoup(content)

    alls = []
    for d in soup.findAll('div', attrs={'class':'a-section a-spacing-none aok-relat
        name = d.find('span', attrs={'class':'zg-text-center-align'})
        n = name.find_all('img', alt=True)
        author = d.find('a', attrs={'class':'a-size-small a-link-child'})
        rating = d.find('span', attrs={'class':'a-icon-alt'})
        users_rated = d.find('a', attrs={'class':'a-size-small a-link-normal'})
        price = d.find('span', attrs={'class':'p13n-sc-price'})
```

3. Collect the results and analyse (as usual)

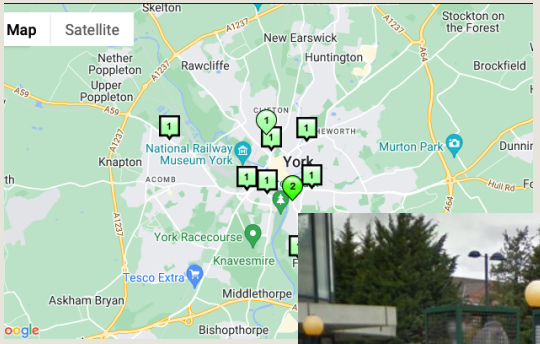
```
results
```

```
[[['Pinch of Nom Quick & Easy: 100 Delicious, Slimming Recipes',
   'Kay Featherstone',
   '4.9 out of 5 stars',
   '11,379',
   '£10.00'],
  ['The Boy, The Mole, The Fox and The Horse',
   'Charlie Mackesy',
   '4.9 out of 5 stars',
   '45,830',
   '£9.00'],
```

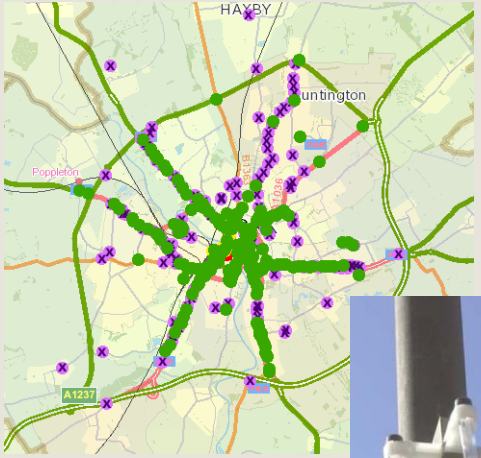
Just because you can scrape it, doesn’t mean you should !

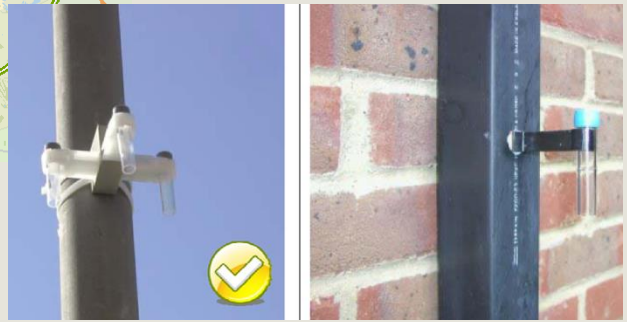
- Are you Collecting personal data (GDPR violation)
- Are there any privacy concerns for their website and/or their clients?
- Do you have the right to publish your analysis or product (e.g., an app based on National Rail)
- Is there an API or fee you neglect to respect?
- Is the organisation willing to share this data?

Existing External Sources Example - Sensor deployment



High Cost
High temporal resolution
Low spatial resolution





Low Cost
Low temporal resolution
Low spatial resolution

Sensor deployment

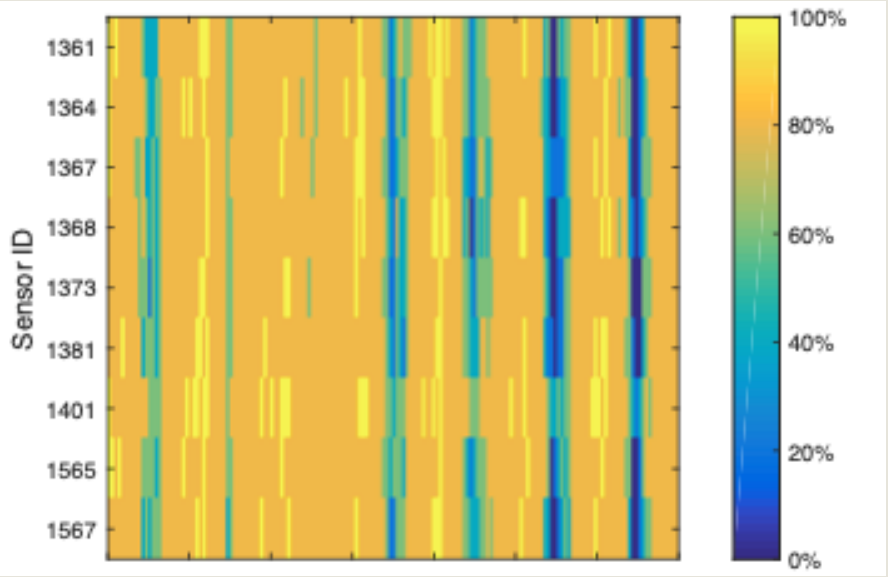
Monitoring (20s sampling frequency)

1. Nitrogen dioxide (NO2),
2. Ground ozone (O3),
3. Nitrogen oxide (NO),
4. Temperature (T),
5. Humidity (H),
6. V=Volatile organic compound (VOC),
7. Dust
8. Noise Level.

Specification

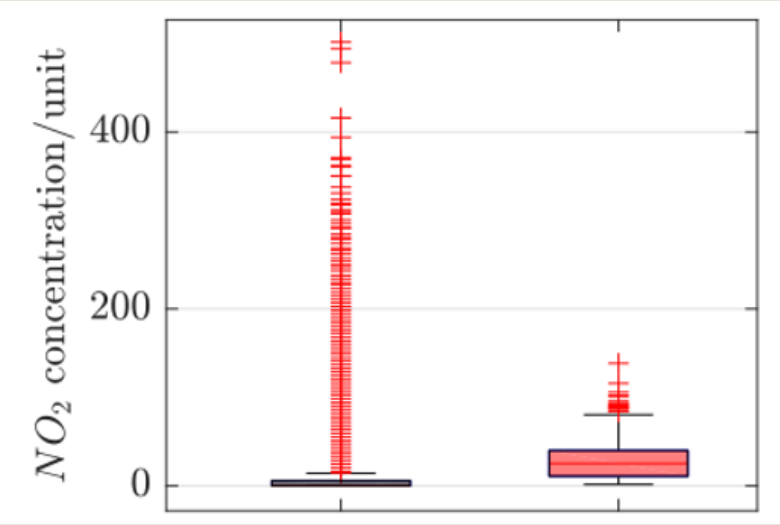
1. £3000/unit
2. ELM sensor from Perkin Elmer
3. Battery or main powered
4. 18 months lifetime
5. GSM communication
6. Cloud storage
7. Temporal onboard storage

Issues – Missing Values



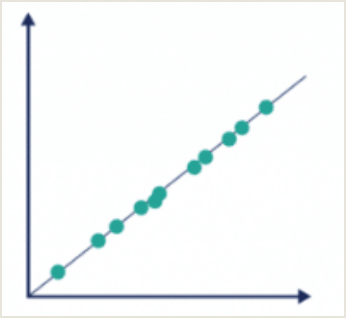
- Nine sensors display an identical pattern of data gaps.
- The period during which the data gap was observed is correlated with the University Open Day event, when the campus was operating at high capacity.
- The GSM communication was therefore constrained by limited bandwidth, resulting in intermittent data transmission and subsequent gaps.

Issues – Data Spikes

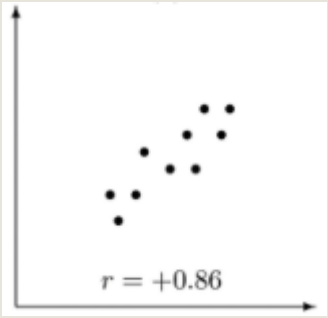


- The ELM sensors exhibit more outliers than the reference sensor.
- The ELM sensors have a sampling frequency of 20 seconds, whereas the reference sensor reports hourly averaged values.
- The spike was likely caused by a diesel bus idling near the ELM sensor.

Issues – Expectation vs Reality

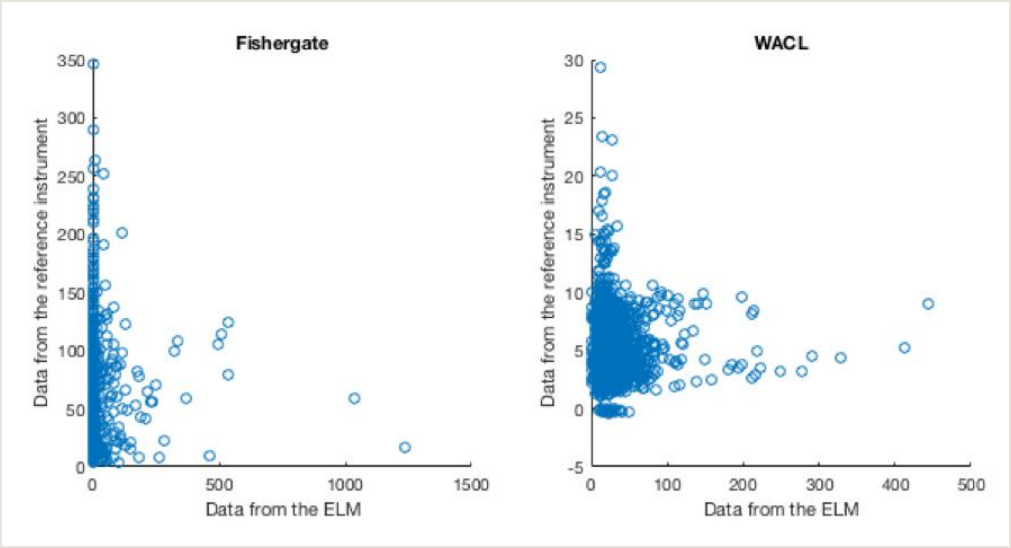


Perfect



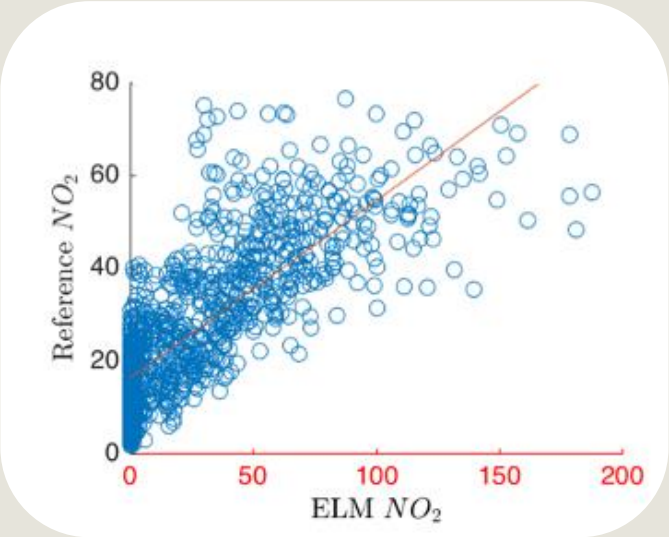
Ideal

The Expectation

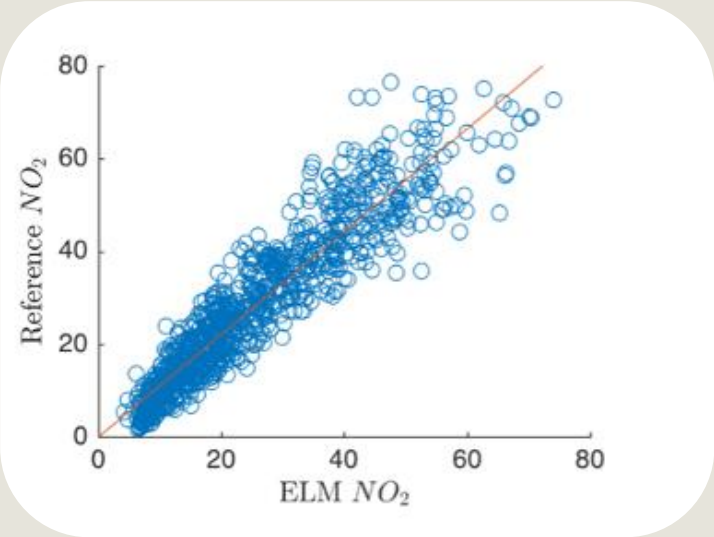


The Reality

Find the root cause and corrected it.



Before



After

Existing External Sources Example - Survey

Please rate your level of satisfaction with your experience at the airport:

	Very Dissatisfied	Dissatisfied	Neutral	Satisfied	Very Satisfied
The overall experience at the airport	◯	◯	◯	◯	◯
Cleanliness & maintenance of airport facilities	◯	◯	◯	◯	◯
Efficiency of check-in & screening process	◯	◯	◯	◯	◯
Clarity of signage & wayfinding	◯	◯	◯	◯	◯
Comfort at the waiting area & lounge	◯	◯	◯	◯	◯

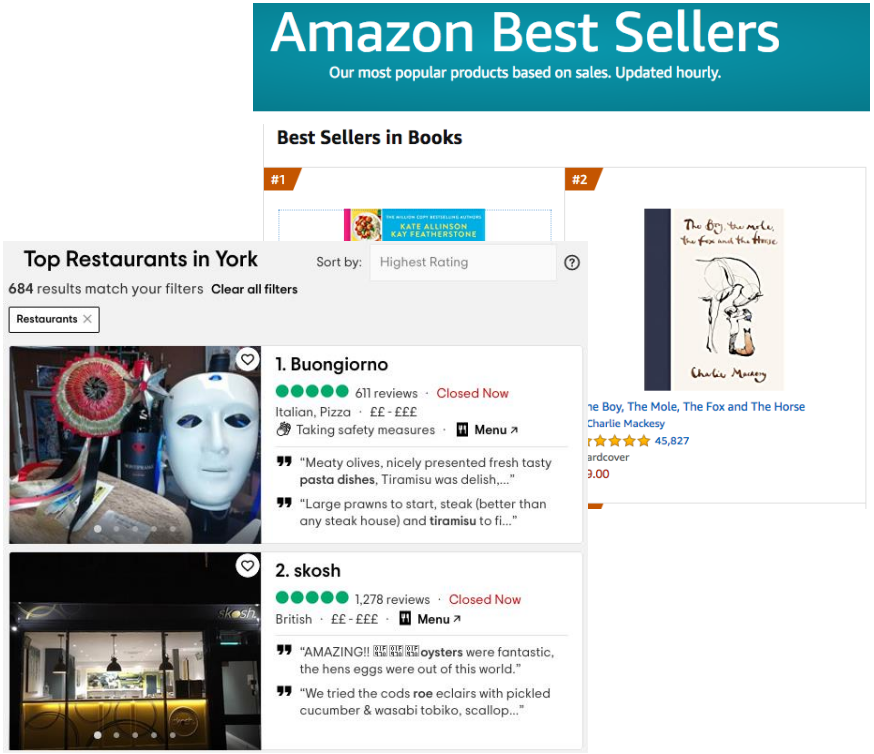
- A structured method of collecting data from individuals
- Enables collection of data directly from people
- Standardised questions allow statistical analysis
- Useful for measuring attitudes and perceptions that cannot be captured by sensors or automated systems

Pitfalls in Survey

- **Sampling bias -** Surveying only university students to estimate national voting behaviour.
- **Low response rates -** Sending a survey to 100 people but receiving only 3 responses, which won't have statistic significance.
- **Poor question design -** "Do you agree that the new system is efficient and user friendly?" This is double barrelled because it asks about two things at once.
- **Ethical and privacy concerns -** Collecting identifiable health data without clear consent or secure storage.
- **Survey fatigue -** A 40 minute questionnaire with repetitive scale items. Respondents may rush or select the same option repeatedly.

External Sources Need Collection Effort

- Available from external sources but not in ideal/standardised formats
- Facing uncertainties, knowledge gap, and practical limitations (sensor deployment)
- Getting access requires special processing
- Ethical and privacy concerns.



Summary

- Data may be classified as structured, semi structured, or unstructured
- Data can be collected from internal and external sources, each with their advantage and disadvantages
- Data collection from external sources can add value but often requires more time and effort to obtain and use effectively

Further Readings.

- ImageNet
- CSV vs JSON vs XML – The Best Comparison Guide 2026
- Data Collection Methods | Primary and Secondary Data
- Web Scraping with Python: Collecting More Data from the Modern Web
- Is Web Scraping Legal?