Data is collected from information

Pipeline:

- Planning
- Gathering data – many sources (websites, user surveys, sensors, legacy databases)
- Processing – cleaning (80% of the time), aligning, integrating (computers are simple, people are not)
- Analysing – statistics, machine learning, exploring (Number crunching, discovering patterns)
- Presenting – visualisations, communication, actionable (message & audience)
- Preserving – storing, management, re-use (data storage & indexing, recording processes – provenance)

Data:

- Structured:
    - Tables, organised, observations
    - Row is an instance; column -> attribute
    - Easier for ML to work with

4 special types of data to look for:

1. Temporal – time series
2. Geographic – spatial

Qualitative:

- Quality, label, trait
- Categorical
- Limited mathematical functions

- E.g. Fav colour, gender

Data sources:

1. Files – texts or binary, open or proprietary, tabulated data (csv, tsv, db), other text data (Json, Html, KML)
2. Databases
3. The Internet
4. Open Data – public data, shared and freely available

03/10/2024

Metadata comes from

- Simple – EXIF,
- Structured – Adhering to a standard
- Professional – created by librarian
- Crowdsourced – hashtags, comments

Assignment:

CSC1143 – visualisation

Any dataset can be used except

Check specification file

Marking criteria – 30% dataset; Visualisation 50%; Report 20%

5 good things to know about data visualisation:

1. Pie charts – never a good idea. It looks good but not very technical when there is loads of information. Only use it for parts of a whole. (1005 divided into groups). No more than 5 slices. Never use 3d.

24/10/2024

Chart types: - CHART

1. Categorical: Comparing categories
    a. Comparison:
        i. Bar graph
        ii. Dot plot
        iii. Circle packing
        iv. Polar/Radar/Spider chart
    b. Distribution:
        i. Box-and-whisker plot
        ii. Histogram
        iii. Word cloud (not very accurate)

2. Hierarchical: Charting part-to-whole relationships
    a. Part-to-whole:
        i. Pie charts
        ii. Waffle charts
        iii. Stacked bar chart
        iv. Tree map
        v. Venn diagram
    b. Hierarchies:

        i. Dendrogram: clusters

3. Relational: Graphing relationships to explore correlations and connections

    a. Connections:

        i. Scatter plot

        ii. Bubble plot

        iii. Heatmap

        iv. Matrix chart

        v. Sankey diagram

4. Temporal: Showing trends/activities over time

    a. trends

        i. Line chart

        ii. Area chart

        iii. Stream graph

    b. Activities:

        i. Gantt chart

5. Spatial: Mapping

    a. Overlays

        i. Choropleth

        ii. Isarithmic

        iii. Proportional symbol

    b. Distortion:

i. Area cartogram

ii. Dorling cartogram

07/12

TILE: Finding data

Scraping	Crawling
to extract data from semi-structured sources (e.g., webpages).	traversing the web via links in <a> tags to gather data via scraping.

The general process of **Scraping** is as follows:

- Have a plan (how to identify the data items on the page)
- Request webpage (e.g., urlopen, requests)
- Parse HTML (e.g., lxml, beautifulsoup)
- Store data (e.g., as list or dict)
- Format as required (e.g., CSV, json, dataframe, sql)

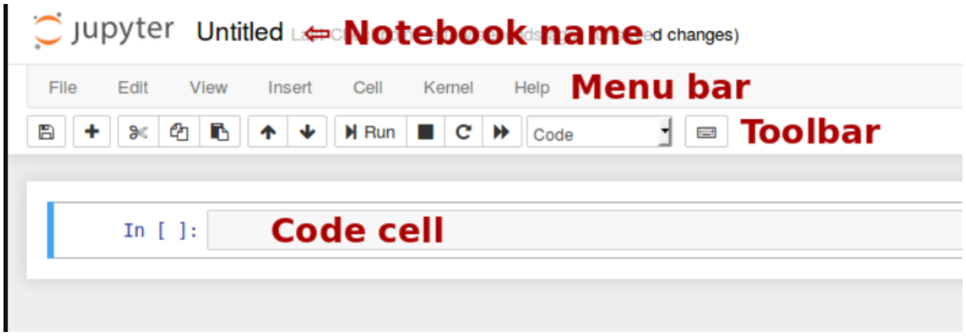
You will almost certainly need to clean the data as scraping can be very prone to introducing errors and artefacts.

Good practices for scarping:

- Check site's terms and conditions
- Make sure you allocate your requests properly and in order so that, you don't hammer the site's server causing denial of service attack.
- Scarper's break - sites change their layout all the time. If that happens, be prepared to rewrite your code.
- Web pages can be messy, so you might need to tidy up the data by hand after you've collected it.

Python libraries to help with scraping

- **requests** - downloading the page
- **BeautifulSoup** - parsing the HTML into an object to search and manipulate
- **Scrapy -** Writing a spider to crawl a site and extract data



APIs are like bridges that allow different software systems to communicate and share data or features with each other. Forms of APIs include:

- REST (REpresentational State Transfer) - more common in recent years
- SOAP (Simple Object Access Protocol) - more powerful
- JSON (JavaScript Object Notation) - used when returning results from REST calls.

What are the advantages and disadvantages of providing or accessing data via an online REST API?

Advantages	Disadvantages
REST APIs are based on the same standards used for the web. They are **highly interoperable and can easily interact.**	the design of a REST API can be more complex than other APIs
Flexibility- REST APIs can communicate using any data format	REST APIs can have slightly lower performance than other APIs
Security - REST APIs typically use authentication via access tokens, Tokens are much more difficult to crack as they are unique.	All changes to your REST API must be executed on the web and only on the web. You must always connect to make the slightest change.
REST APIs are **highly scalable**	
REST APIs are simpler and easier to use than other APIs.	

TILE: Data Quality & Cleaning

High-quality data is free from both errors and artefacts.

- **Error**: data that is missing or lost due to the capture process and cannot be recovered.
- **Artefact**: something that has been introduced into the dataset during the gathering, processing, integration or cleaning activities.

Where do we face issues with data quality or cleaning?

1. **Data gathering & Data Retrieval**
    a. Manual entry errors like - (duc instead of dcu)
    b. Poor survey or interface design (asking the user their favourite colour but the only options available are red, green, blue)
    c. No standard format - (DCU or Dublin City University)
    d. Source data not understood

    e. *Solutions* - build in integrity check, Process management - reward accurate human data entry. Cleaning focus (duplicate removal, merge/purge), Diagnostic focus (automated detection of glitches).

2. **Data delivery (Data Processing):**
    a. Destroying or mutilating information by inappropriate pre-processingInappropriate aggregation - Nulls converted to default values
    b. Loss of data
    c. *Solutions* - Build reliable transmission protocols, verification (checksums), interface agreements (data quality commitment)

3. **Data Storage (preserving)**
    a. Format conversion errors (string/float)
    b. No meta data
    c. Transmission errors, corruption

4. **Data Integration**
    a. Combining data sets - (Heterogeneous data : no common key, different field formats, Different definitions : What is a customer, an account, a family, Time synchronization : Does the data relate to the same time periods?, Legacy data : IMS, spreadsheets,Sociological factors : Reluctance to share – loss of power)
    b. *Solutions* - Have metadata

5. **Data Analysis**
    a. Scale and performance
    b. Insufficient domain expertise

Data quality is defined as the degree to which data meets a company’s expectations of accuracy, validity, completeness, and consistency.

Measures of **Data Quality:**
- Accuracy : The data was recorded correctly.
- Completeness : All relevant data was recorded.
- Uniqueness : Entities are recorded once.
- Timeliness : The data is recent or kept up to date. Date published vs Data captured ...
- Consistency : The data agrees with itself (internal).
- Credibility : The data comes from a recognised (or official) source.

Ways to clean data:
- Implement process mandates - fixing the human problem by including schema to maintain a format.
- Custom tools written in a General Purpose language (hack a script) Good for one-off quick fixes (cleaning that only happens once)

Suggestions:

- Missing values, records or variables - are empty cells no value (0) or no measurement (null)? How should they be handled?
- Erroneous values - typos or values that are clearly out of place (gender value in age column
- Inconsistencies - capitalisation, units of measurement
- Duplicate records
- Out of date - e.g., age will have changed
- Leading or trailing spaces! Windows or Linux end of line characters
- Format of dates - DD/MM/YYYY, MM/DD/YYYY, ?? Excel based or Unix based
- “Sanity checks” - look for extreme values or outliers, count how many record