# NPA Data Science

# 5: Data capture and modelling

## Topic Outcomes

- Explain techniques for data capture,
- Explain techniques for data cleaning and transformation, including data modelling

## Data Capture

Data capture is the action or process of gathering data, especially from an automatic device, control system, or sensor.

There are numerous methods for data capture that can be utilised to capture key information from surveys, invoices, claim forms, unstructured documents and other document types. Thanks to innovative technology, data capture processes can be automated to reduce manual data entry and increase data accuracy. Although there are many methods of capturing data automatically, many businesses and organisations prefer to capture it manually. The use of paper forms for example are still prevalent. However in the context of data science and the internet of things, it is the prevalence and growth of smart devices that offer the most scope for innovative methods of data capture across many sectors of society. The benefits of using electronic data capture include a reduction in time and increased efficiency, as well as the ability to capture a far larger volume of data. Automatic data capture also removes the need for human involvement.

### Optical Character Recognition (OCR)

OCR technology is used to capture data from structured documents, usually those that have been word processed. The software works by converting documents into machine readable files, once this has happened you can search by keywords contained within it, great for files containing large amounts of data.

### Intelligent Character Recognition (ICR)

ICR technology is an adapted version of OCR, the difference being that ICR is able to read handwritten text and convert it into computer readable information. Often used for unstructured documents such as letters, unstructured documents and other handwritten business correspondence.

### Automatic Data Capture

Once documents have been OCR or ICR scanned, automatic data capture software can identify and extract key information from forms. This is becoming extremely popular for businesses looking to automate processing tasks, such as invoicing, purchasing and claims processing. The software is trained to look for specific types of information, such as

reference numbers, names and addresses. Extracted information is output into a preferred format for import into existing systems, commonly used are csv, excel or html.

### Paperless Forms

Paperless forms allow information to be captured whilst out in the field, transforming the way data comes in and out of a business. Data is captured using a mobile device and can be transferred straight into office systems, with no paper processing involved or delays.

### Barcode Technology

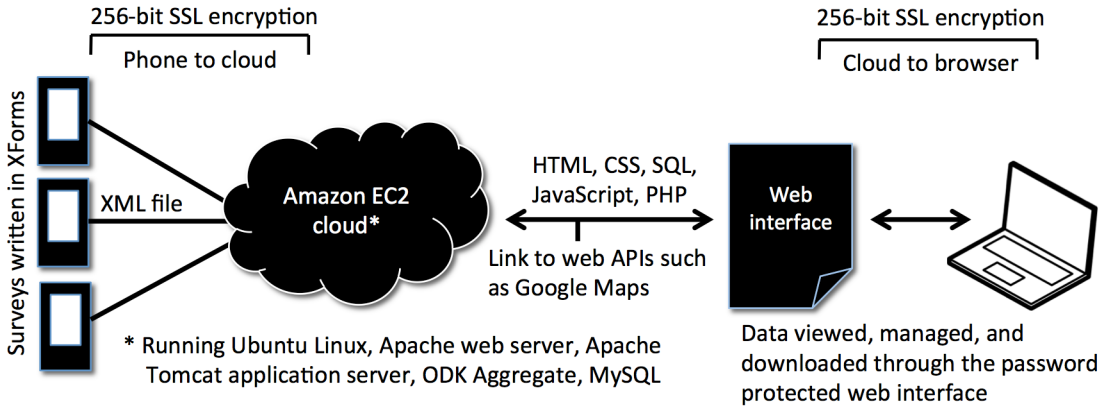
Barcode technology is a data capture method that allows metadata such as customer name, address and contact numbers to be pre populated into barcode format prior to forms being sent. This form of data capture significantly reduces manual data entry requirements upon return.

### Double Blind Data Entry

Double blind data entry is a form of manual data entry using two operators and validation software to increase accuracy. The software will flag up any discrepancies between both sets of data and errors must be corrected before they can move.

### Data Capture Examples

Bespoke systems are often developed for specific business needs and projects. One such example is the LINKS system, for health programmes shown in Figure 1 below, with the data flow from the point of entry (cellular phones/tablets), to encrypted transmission to the cloud, to access and management of the data through a web interface.




The LINKS system has been deployed in many countries as part of global health programmes, with between 10-200 data collectors and between 6000 to 1,000,000 records collected. There are significant benefits from the use of open source technologies in the system, alongside a reduction in time, effort and cost. A non-cloud-based system requires manual synchronisation of data using local laptop computers, adding time and equipment. In contrast, a cloud-based system automatically synchronises the data directly from the smartphones whenever connected to the internet, allowing data managers to identify and communicate issues with the field team during the collection of data. This allows data from

all project sites to flow from collection to implementation more rapidly. In the case of the GTMP, results take, on average, three days from the end of data collection to be included in public programmatic tools and for program implementation planning.

**Big Data Capture in Business**

Businesses often have what is called a data capture strategy, which outlines the methods for capturing and managing data. Companies that effectively create and implement big data strategies stand to gain a competitive advantage.

Four strategies for capturing big data include performance management, data exploration, social analytics and decision science. Which is adopted depends on data type and business objectives, and whether there is transactional or non-transactional data involved, and measurement or experimentation.

Performance management involves understanding the meaning of big data in company databases using predetermined queries and multidimensional analysis. The data used for this analysis are transactional, for example, years of customer purchasing activity, and inventory levels and turnover

Data exploration makes heavy use of statistics to experiment and get answers to questions that managers might not have thought of previously. This approach leverages predictive modelling techniques to predict user behaviour based on their previous business transactions and preferences. Cluster analysis can be used to segment customers into groups based on similar attributes that may not have been on analysts' radar screens. Once these groups are discovered, managers can perform targeted actions such as customising marketing messages, upgrading service, and cross/up-selling to each unique group.

Social analytics measure the vast amount of non-transactional data that exists today. Much of this data exists on social media platforms, such as conversations and reviews on Facebook, Twitter, and Yelp. Social analytics measure three broad categories: awareness, engagement, and word-of-mouth or reach.3 Awareness looks at the exposure or mentions of social content and often involves metrics such as the number of video views and the number of followers or community members. Engagement measures the level of activity and interaction among platform members, such as the frequency of user-generated content.

Decision science involves experiments and analysis of non-transactional data, such as consumer-generated product ideas and product reviews, to improve the decision-making process. Unlike social analysers who focus on social analytics to measure known objectives, decision scientists explore social big data as a way to conduct "field research" and to test hypotheses. Crowdsourcing, including idea generation and polling, enables companies to pose questions to the community about its products and brands. Decision scientists, in conjunction with community feedback, determine the value, validity, feasibility and fit of these ideas and eventually report on if/how they plan to put these ideas in action.

For example, the My Starbucks Idea program enables consumers to share, vote, and submit ideas regarding Starbucks products, customer experience, and community involvement. Over 100,000 ideas have been collected to date. Starbucks has an “Ideas in Action" section to discuss where ideas sit in the review process.

However, when implementing a data capture strategy, there are several challenges which must be overcome. Aside from the fundamental issues of capturing data including missing pages, blank fields, spelling mistakes and incorrect variables, issues with the setup of the forms can lead to collection of inaccurate data. Free text fields in web forms require manual, real-time validation and should be avoided where possible as they allow for numerous variations in their responses. Questionable data quality can render the data as null and void, or alternatively result in expensive cleansing campaigns.

## Data Cleaning

You may be unaware of the messy nature of most data and the time consuming nature of data cleaning and transformation, prior to analysis. Data pre-processing is the first (and arguably most important) step toward building a working machine learning model. It’s critical! If your data hasn’t been cleaned and pre-processed, your model does not work. It’s that simple.

Data cleaning involves different techniques based on the problem and the data type. Different methods can be applied with each having its own trade-offs. Overall, incorrect data is either removed, corrected, or imputed. Here are some of the problems that data cleaning and pre-processing try to rectify.

### Irrelevant data

Irrelevant data are those that are not actually needed, and don’t fit under the context of the problem we’re trying to solve. For example, if we were analysing data about the general health of the population, the phone number wouldn’t be necessary — column-wise. Similarly, if you were interested in only one particular country, you wouldn’t want to include all other countries. Or, study only those patients who went to the surgery, we wouldn’t include everyone — row-wise. Only if you are sure that a piece of data is unimportant, you may drop it. Otherwise, explore the correlation matrix between feature variables. And even though you noticed no correlation, you should ask someone who is a domain expert. You never know, a feature that seems irrelevant, could be very relevant from a domain perspective such as a clinical perspective.

### Duplicates

Duplicates are data points that are repeated in your dataset. It often happens when, for example, data types are combined from different sources. The user may hit the submit button twice thinking the form wasn’t actually submitted. A request for online bookings was submitted twice, correcting wrong information that was entered accidentally the first time. A common symptom is when two users have the same identity number. Or, the same article was scrapped twice. And therefore, they simply should be removed.

**Type conversion**

Make sure numbers are stored as numerical data types. A date should be stored as a date object, or a Unix timestamp (number of seconds), and so on. Categorical values can be converted into and from numbers if needed. This can be spotted quickly by taking a peek over the data types of each column in the summary (we've discussed above). A word of caution is that the values that can't be converted to the specified type should be converted to NA value (or any), with a warning being displayed. This indicates the value is incorrect and must be fixed.

**Syntax errors**

Remove white spaces: Extra white spaces at the beginning or the end of a string should be removed.

"  hello world  " => "hello world

Pad strings: Strings can be padded with spaces or other characters to a certain width. For example, some numerical codes are often represented with prepending zeros to ensure they always have the same number of digits.

313 => 000313 (6 digits)

Fix typos: Strings can be entered in many different ways, and no wonder, can have mistakes.

Gender

m

Male

fem.

FemalE

Femle

This categorical variable is considered to have 5 different classes, and not 2 as expected: male and female since each value is different.

A bar plot is useful to visualise all the unique values. One can notice some values are different but do mean the same thing i.e. "information technology" and "IT". Or, perhaps, the difference is just in the capitalisation i.e. "other" and "Other".

Therefore, our duty is to recognise from the above data whether each value is male or female. How can we do that?.

The first solution is to manually map each value to either "male" or "female".

```
dataframe['gender'].map({'m': 'male', fem.': 'female', ...})
```

The second solution is to use pattern matching. For example, we can look for the occurrence of m or M in the gender at the beginning of the string.

re.sub(r"\^m\$", 'Male', 'male', flags=re.IGNORECASE)

The third solution is to use fuzzy matching: An algorithm that identifies the distance between the expected string(s) and each of the given ones. Its basic implementation counts how many operations are needed to turn one string into another.

Gender male female

m 3 5

Male 1 3

fem. 5 3

FemalE 3 2

Femle 3 1

Furthermore, if you have a variable like a city name, where you suspect typos or similar strings should be treated the same. For example, “lisbon” can be entered as “lisboa”, “lisbona”, “Lisbon”, etc.

City Distance from "Lisbon"

lisbon 0

lisboa 1

Lisbon 1

lisbona 2

london 3

...

If so, then we should replace all values that mean the same thing to one unique value. In this case, replace the first 4 strings with “lisbon”.

Watch out for values like “0”, “Not Applicable”, “NA”, “None”, “Null”, or “INF”, they might mean the same thing: The value is missing.

**Standardise**

Our duty is to not only recognise the typos but also put each value in the same standardised format. For strings, make sure all values are either in lower or upper case. For numerical values, make sure all values have a certain measurement unit. The height, for example, can be in meters and centimetres. The difference of 1 meter is considered the same as the difference of 1 centimetre. So, the task here is to convert the heights to one single unit. For dates, the USA version is not the same as the European version.

Recording the date as a timestamp (a number of milliseconds) is not the same as recording the date as a date object.

**Scaling / Transformation**

Scaling means to transform your data so that it fits within a specific scale, such as 0–100 or 0–1. For example, exam scores of a student can be re-scaled to be percentages (0–100) instead of GPA (0–5). It can also help in making certain types of data easier to plot. For example, we might want to reduce skewness to assist in plotting (when having so many outliers). The most commonly used functions are log, square root, and inverse.

Scaling can also take place on data that has different measurement units. Student scores on different exams say, SAT and ACT, can't be compared since these two exams are on a different scale. The difference of 1 SAT score is considered the same as the difference of 1 ACT score. In this case, we need re-scale SAT and ACT scores to take numbers, say, between 0–1. By scaling, we can plot and compare different scores.

**Normalisation**

While normalisation also rescales the values into a range of 0–1, the intention here is to transform the data so that it is normally distributed. Why? In most cases, we normalise the data if we're going to be using statistical methods that rely on normally distributed data. How? One can use the log function, or any of these statistical methods.

**Missing values**

Given the fact the missing values are unavoidable, it leaves us with the question of what to do when we encounter them. Ignoring the missing data is the same as digging holes in a boat; It will sink. Missing values are not the same as default values. For instance, zero can be interpreted as either missing or default, but not both. Missing values are not "unknown". Research conducted where some people didn't remember whether they have been bullied or not at the school, should be treated and labelled as unknown and not missing.

There are three, or perhaps more, ways to deal with them.

1. Drop.

If the missing values in a column rarely happen and occur at random, then the easiest and most forward solution is to drop observations (rows) that have missing values. If most of the column's values are missing, and occur at random, then a typical decision is to drop the whole column. This is particularly useful when doing statistical analysis, since filling in the missing values may yield unexpected or biased results.

2. Impute.

It means to calculate the missing value based on other observations. There are quite a lot of methods to do that. For example, using statistical values like mean, median. However, none of these guarantees unbiased data, especially if there are many missing values. Mean is most useful when the original data is not skewed, while the median is more

robust, not sensitive to outliers, and thus used when data is skewed. In a normally distributed data, one can get all the values that are within 2 standard deviations from the mean. Next, fill in the missing values by generating random numbers between (mean — 2 * std) & (mean + 2 * std). Also using a linear regression. Based on the existing data, one can calculate the best fit line between two variables, say, house price vs. size m². It is worth mentioning that linear regression models are sensitive to outliers. Finally, hot-deck: Copying values from other similar records. This is only useful if you have enough available data. And, it can be applied to numerical and categorical data.

3. Flag

Some argue that filling in the missing values leads to a loss in information, no matter what imputation method we used. That's because saying that the data is missing is informative in itself, and the algorithm should know about it. Otherwise, we're just reinforcing the pattern already existing by other features. This is particularly important when the missing data doesn't happen at random. Take for example a conducted survey where most people from a specific race refuse to answer a certain question. Missing numeric data can be filled in with say, 0, but these zeros must be ignored when calculating any statistical value or plotting the distribution. While categorical data can be filled in with say, "Missing": A new category which tells that this piece of data is missing.

Every time we drop or impute values we are losing information. So, flagging might come to the rescue.

**Outliers**

They are values that are significantly different from all other observations. Any data value that lies more than (1.5 * IQR) away from the Q1 and Q3 quartiles is considered an outlier. Outliers are innocent until proven guilty. With that being said, they should not be removed unless there is a good reason for that. For example, one can notice some weird, suspicious values that are unlikely to happen, and so decides to remove them. Though, they are worth investigating before removing. It is also worth mentioning that some models, like linear regression, are very sensitive to outliers. In other words, outliers might throw the model off from where most of the data lie.

**In-record and cross-datasets errors**

These errors result from having two or more values in the same row or across datasets that contradict each other. For example, if we have a dataset about the cost of living in cities. The total column must be equivalent to the sum of rent, transport, and food.

## Data Modelling

Data modelling is the process of creating a visual representation of either a whole information system or parts of it to communicate connections between data points and structures. The goal is to illustrate the types of data used and stored within the system, the relationships among these data types, the ways the data can be grouped and organised and its formats and attributes.

Data models are built around business needs. Rules and requirements are defined upfront through feedback from business stakeholders so they can be incorporated into the design of a new system or adapted in the iteration of an existing one.

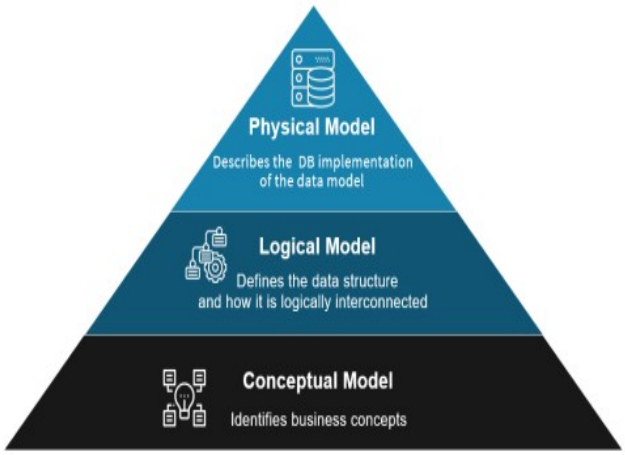
Data can be modelled at various levels of abstraction. The process begins by collecting information about business requirements from stakeholders and end users. These business rules are then translated into data structures to formulate a concrete database design. A data model can be compared to a roadmap, an architect's blueprint or any formal diagram that facilitates a deeper understanding of what is being designed.

Data modelling employs standardised schemas and formal techniques. This provides a common, consistent, and predictable way of defining and managing data resources across an organisation, or even beyond.

Benefits of Data Modelling for Organisations are:

- Higher quality software development.
- Reduced costs.
- Faster time to market.
- Clear understanding of scope, vocabulary, and other development elements.
- Better application and database performance.
- High quality documentation.
- Fewer errors in software.
- Fewer data errors across organisational systems.
- Better risk management.

Like any design process, database and information system design begins at a high level of abstraction and becomes increasingly more concrete and specific. Data models can generally be divided into three categories, which vary according to their degree of abstraction. The process will start with a conceptual model, progress to a logical model and conclude with a physical model as shown in the figure below.



**Conceptual data models.** They are also referred to as domain models and offer a big-picture view of what the system will contain, how it will be organised, and which business rules are involved. Conceptual models are usually created as part of the process of gathering initial project requirements.

**Logical data models.** They are less abstract and provide greater detail about the concepts and relationships in the domain under consideration. One of several formal data modelling notation systems is followed. These indicate data attributes, such as data types and their corresponding lengths, and show the relationships among entities. Logical data models don't specify any technical system requirements. This stage is frequently omitted in agile or DevOps practices.

**Physical data models.** They provide a schema for how the data will be physically stored within a database. As such, they're the least abstract of all. They offer a finalised design that can be implemented as a relational database, including associative tables that illustrate the relationships among entities as well as the primary keys and foreign keys that will be used to maintain those relationships.

Data modelling has evolved alongside database management systems, with model types increasing in complexity as businesses' data storage needs have grown. Here are several model types:

**Hierarchical data models** represent one-to-many relationships in a treelike format. In this type of model, each record has a single root or parent which maps to one or more child tables. This model was implemented in the IBM Information Management System (IMS), which was introduced in 1966 and rapidly found widespread use, especially in banking. Though this approach is less efficient than more recently developed database models, it's still used in Extensible Markup Language (XML) systems and geographic information systems (GISs).

**Relational data models** were initially proposed by IBM researcher E.F. Codd in 1970. They are still implemented today in the many different relational databases commonly used in enterprise computing. Relational data modelling doesn't require a detailed understanding of the physical properties of the data storage being used. In it, data segments are explicitly joined through the use of tables, reducing database complexity.

**Entity-relationship (ER)** data models use formal diagrams to represent the relationships between entities in a database. Several ER modelling tools are used by data architects to create visual maps that convey database design objectives.

**Object-oriented data models** gained traction as object-oriented programming and it became popular in the mid-1990s. The "objects" involved are abstractions of real-world entities. Objects are grouped in class hierarchies, and have associated features. Object-oriented databases can incorporate tables, but can also support more complex data relationships. This approach is employed in multimedia and hypertext databases as well as other use cases.

**Dimensional data models** were developed by Ralph Kimball, and they were designed to optimise data retrieval speeds for analytic purposes in a data warehouse. While relational and ER models emphasise efficient storage, dimensional models increase redundancy in order to make it easier to locate information for reporting and retrieval. This modelling is typically used across OLAP systems.

### The Process of Data Modelling

All approaches provide formalised workflows that include a sequence of tasks to be performed in an iterative manner. Those workflows generally look like this:

1. **Identify the entities.** The process of data modelling begins with the identification of the things, events or concepts that are represented in the data set that is to be modelled. Each entity should be cohesive and logically discrete from all others.
2. **Identify key properties of each entity**. Each entity type can be differentiated from all others because it has one or more unique properties, called attributes. For instance, an entity called “customer” might possess such attributes as a first name, last name, telephone number and salutation, while an entity called “address” might include a street name and number, a city, state, country, and zip code.
3. **Identify relationships among entities**. The earliest draft of a data model will specify the nature of the relationships each entity has with the others. In the above example, each customer “lives at” an address. If that model were expanded to include an entity called “orders,” each order would be shipped to and billed to an address as well. These relationships are usually documented via unified modelling language (UML).
4. **Map attributes to entities.** This will ensure the model reflects how the business will use the data. Several formal data modelling patterns are in widespread use. Object-oriented developers often apply analysis patterns or design patterns, while stakeholders from other business domains may turn to other patterns.
5. **Assign keys as needed,** and decide on a degree of normalisation that balances the need to reduce redundancy with performance requirements. Normalisation is a technique for organising data models (and the databases they represent) in which numerical identifiers, called keys, are assigned to groups of data to represent relationships between them without repeating the data. For instance, if customers are each assigned a key, that key can be linked to both their address and their order history without having to repeat this information in the table of customer names. Normalisation tends to reduce the amount of storage space a database will require, but it can at cost to query performance.
6. **Finalise and validate the data model.** Data modelling is an iterative process that should be repeated and refined as business needs change.

Finally, it is useful to be aware that with the increasing use of NoSQL and the variety of systems under this umbrella, some question the relevance of traditional notions of model and modelling to this area. NoSQL systems have gained in popularity for many reasons, including the flexibility they provide in organising data, as they relax the rigidity provided by the relational model and by the other structured models. This flexibility and the heterogeneity that has emerged in the area have led to a little use of traditional modelling techniques, as opposed to what has happened with databases for decades. However, traditional notions related to data modelling can still be useful in this context as well (Atzeni et al. 2020). And others actually argue that the future of data is through SQL as the universal interface for data analysis (Kulkarni 2019).

## References


Atzeni, P. et al. 2020. Data modelling in the NoSQL World. *Computer Standards & Interfaces.* 67. Available: https://www.sciencedirect.com/science/article/abs/pii/S0920548916301180

Clear Data. Data Capture Methods. Available at: https://www.ukdataentry.com/data-capture-solutions/data-capture-methods/

Elgabry, O. 2019. The Ultimate Guide to Data Cleaning. Towards Data Science. Available at: https://towardsdatascience.com/the-ultimate-guide-to-data-cleaning-3969843991d4

FintechOS. Evolutive Data Model. Available: https://fintechos.com/documentation/Studio/20.1/UserGuide/Content/Evolutive%20Data%20Model/Evolutive%20Data%20Model.htm

Kulkarni, A. 2019. NoSQL vs SQL: The Future of Data. Timescale. Available at: https://blog.timescale.com/blog/why-sql-beating-nosql-what-this-means-for-future-of-data-time-series-database-348b777b847a/

IBM Cloud Education. 2020. What is Data Modelling? Available at: https://www.ibm.com/cloud/learn/data-modeling

Parise, S. et al. 2012. Four strategies to capture and create value from big data. Ivey Business Journal: Improving the Practice of Management. Available at: https://iveybusinessjournal.com/publication/four-strategies-to-capture-and-create-value-from-big-data/

Pavluck, A. et al. 2014. Electronic Data Capture Tools for Global Health Programs: Evolution of LINKS, an Android-Web-Based System. PLOS Neglected Tropical Diseases. Available at: https://doi.org/10.1371/journal.pntd.0002654