MACHINE LEARNING 101

Patrick Boily[1,2,3], Jen Schellinck[2,4,5]

Abstract
In October 2012, the *Harvard Business Review* published an article calling data science the "sexiest job of the 21st century", and declaring data scientists to be a "hybrid of data hacker, analyst, communicator, and trusted adviser" [11]. Would-be data scientists are usually introduced to the field via machine learning algorithms and applications, which we discuss briefly in this chapter.

Keywords
Machine learning, statistical learning, supervised learning, unsupervised learning, association rules mining, classification, decision trees, clustering, k—means, issues and challenges.

Funding Acknowledgement
Parts of this chapter were funded by Carleton University's Centre for Quantitative Analysis and Decision Support.

[1]Department of Mathematics and Statistics, University of Ottawa, Ottawa, Canada
[2]Data Action Lab, Ottawa, Canada
[3]Idlewyld Analytics and Consulting Services, Wakefield, Canada
[4]Sysabee, Ottawa, Canada
[5]Institute of Cognitive Science, Carleton University, Ottawa, Canada
Email: pboily@uottawa.ca

Contents

1. Introduction

From Data to Wisdom

Data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom.

– attributed to Cliff Stoll, *Nothing to Hide: Privacy in the 21st Century*, 2006

One of the challenges of working in the **data science** (DS), **machine learning** (ML) and **artificial intelligence** (AI) fields is that nearly all quantitative work can be described with some combination of the terms DS/ML/AI (very often to a ridiculous extent).

Robinson [45] suggests that their relationships follow an inclusive hierarchical structure:

- in a first stage, **DS** provides "insights" *via* visualization and (manual) inferential analysis;
- in a second stage, **ML** yields "predictions" (or "advice"), while reducing the operator's analytical, inferential and decisional workload (although it is still present to some extent), and
- in the final stage, **AI** removes the need for oversight, allowing for automatic "actions" to be taken by a completely unattended system.

The goals of artificial intelligence are laudable in an academic setting, but in practice, we believe that stakeholders should not seek to abdicate all of their agency in the decision-making process; as such, we follow the lead of various thinkers and suggest further splitting AI into "general AI" (which we will not be pursuing) and "**augmented** intelligence" (which can be seen as ML "on steroids").

With this in mind, our definition of the DS/ML/AI approach is that it consists of quantitative processes (what Hilary Mason has called "the **working intersection** of statistics, engineering, computer science, domain expertise, and "hacking" [55]) that can help users **learn actionable insights** about their situation without completely abdicating their decision-making responsibility.

In this chapter, then we will take a brief look at:

- the **fundamentals** of data science;
- **association rules** mining;
- **supervised learning and classification**, with a focus on **decision trees**;
- **unsupervised learning and clustering**, with a focus on k–**means**, and
- some of the common **issues and challenges** encountered during the data science and machine learning process.

Later chapters will discuss various other aspects of the general data science and machine learning picture.

2. Fundamentals

From Data to Wisdom

We learn from failure, not from success!

– Bram Stoker, *Dracula*

As humans, we learn (at all stages) by first taking in our environment, and then by answering questions about it, testing hypotheses, creating concepts, making predictions, creating categories, and classifying and grouping its various objects and attributes.

In a way, the main concept under DS/ML/AI is to try to teach our machines (and thus, ultimately, ourselves) to gleam insight from data, and how to do this properly and efficiently, free of biases and pre-conceived notions – in other words, **can we design algorithms that can learn?**

In that context, the simplest DS/ML/AI method is **exploring the data** (or a representative sample) to

- provide a summary through basic statistics – mean, mode, histograms, etc.;
- make its multi-dimensional structure evident through data data visualization; and
- look for consistency, considering what is in there and what is missing.

2.1 Learning Types

In the data science context, more sophisticated approaches traditionally fall into a **supervised** or an **unsupervised** learning framework.

Supervised learning is akin to "learning with a teacher." Typical tasks include **classification**, **regression**, **rankings**, and **recommendations**.

In supervised learning, algorithms use **labeled training data** to build (or train) a predictive model (i.e. "students give an answer to each exam question based on what they learned from worked-out examples provided by the teacher/textbook"); each algorithm's performance is evaluated using **test data** for which the label is known but not used in the prediction (i.e. "the teacher provides the correct answers and marks the exam questions using the key".)

Unsupervised learning, on the other hand, is akin to "self-learning by grouping similar exercises together as a study guide." Typical tasks include **clustering**, **association rules discovery**, **link profiling**, and **anomaly detection**. Unsupervised algorithms use **unlabeled data** to find natural patterns in the data (i.e. "the teacher is not involved in the discovery process"); the drawback is that accuracy **cannot be evaluated** with the same degree of satisfaction (i.e. "students might end up with different groupings").

In supervised learning, there are fixed **targets** against which to train the model (such as age categories, or plant species) – the categories (and their number) are known prior to the analysis.

In unsupervised learning, we don't know what the target is, or even if there is one – we are simply looking for **natural groups** in the data (such as junior students who like literature, have longish hair, and know how to cook **vs.** students who are on a sports team and have siblings **vs.** financial professionals with a penchant for superhero movies, craft beer and Hello Kitty backpack **vs.** ...).

Some data science techniques fit into both camps; others can be either supervised or unsupervised, depending on how they are applied, but there are other conceptual approaches, especially for AI tasks:

- **semi-supervised learning** in which some data points have labels but most do not, which often occurs when acquiring data is costly ("the teacher provides worked-out examples and a list of unsolved problems to try out; the students try to find similar groups of unsolved problems and compare them with the solved problems to find close matches"), and
- **reinforcement learning**, where an agent attempts to collect as much (short-term) reward as possible while minimizing (long-term) regret ("embarking on a Ph.D. with an advisor... with the highs and the lows and **maybe** a diploma at the end of the process?").

2.2 Data Science Tasks

Outside of academia, DS/ML/AI methods are only really interesting when they help users ask and answer useful questions. Compare, for instance:

- **Analytics** – "How many clicks did this link get?"
- **Data Science** – "Based on the previous history of clicks on links of this publisher's site, can I predict how many people from Manitoba will read this specific page in the next three hours?" or "Is there a relationship between the history of clicks on links and the number of people from Manitoba who will read this specific page?"
- **Quantitative Methods** – "We have no similar pages whose history could be consulted to make a prediction, but we have reasons to believe that the number of hits will be strongly correlated with the temperature in Winnipeg. Using the weather forecast over the next week, can we predict how many people will access the specific page during that period? "

Data science models are usually **predictive** (not **explanatory**): they show connections, and exploit correlations to make predictions, but they don't reveal why such connections exist.

Quantitative methods, on the other hand, usually assume a certain level of causal understanding based on various **first principles**. That distinction is not always understood properly by clients and consultants alike.

Common data science tasks (with representative questions) include [42]:

- **classification** and **probability estimation** – which undergraduates are likely to succeed at the graduate level?
- **value estimation** – how much is a given client going to spend at a restaurant?
- **similarity matching** – which prospective clients are most similar to a company's establishes best clients?
- **clustering** – do signals from a sensor form natural groups?
- **association rules discovery** – what books are commonly purchased together at an online retailer?
- **profiling** and **behaviour description** – what is the typical cell phone usage of this customer's segment?
- **link prediction** – J. and K. have 20 friends in common: perhaps they'd be great friends?

A classic example is provided by the UCI Machine Learning Repository Mushroom Dataset [14]. Consider **Amanita muscaria** (fly agaric), a specimen of which is shown in Figure 1. Is it **edible**, or **poisonous**?

There is a simple way to get an answer – eat it, wait, and see. If you do not die or get sick upon ingestion, it was **edible**; otherwise it was **poisonous**.

Figure 1. *Amanita muscaria* (fly agaric), in the wild. Does it look dangerous to you?

This test in unappealing for various reasons, however. Apart from the obvious risk of death, we might not learn much from the experiment; it is possible that this specific specimen was poisonous due to some mutation or some other factor (or that you had a pre-existing condition which combined with the fungus to cause you discomfort, etc.), and that fly agaric is actually edible in general, or **vice-versa**.

A predictive model, which would use features of a vast collection of mushroom species and specimens (including whether they were poisonous or edible) could help shed light on the matter: what do poisonous mushrooms have in common? What properties do edible mushrooms share?[1]

For instance, let's say that **Amanita muscaria** has the following features:

- **habitat**: woods;
- **gill size**: narrow;
- **spores**: white;
- **odor**: none;
- **cap color**: red.

We do not know **a priori** whether it is poisonous or edible. Is the available information sufficient to answer the question? Not on its own, no.[2]

But we could use **past data**, with correct **edible** or **poisonous** labels and the **same set of predictors**, to build various supervised **classification** models to attempt to answer the question.

A simple such model, a **decision tree**, is shown on the left in Figure 2.

[1]Note that this is not the same as understanding **why** a mushroom is poisonous or edible – the data alone cannot provide an answer to that question.

[2]A mycologist could perhaps deduce the answer from these features alone, but she would be using her experience with fungi to make a prediction, and so would not be looking at the features in a *vacuum*.

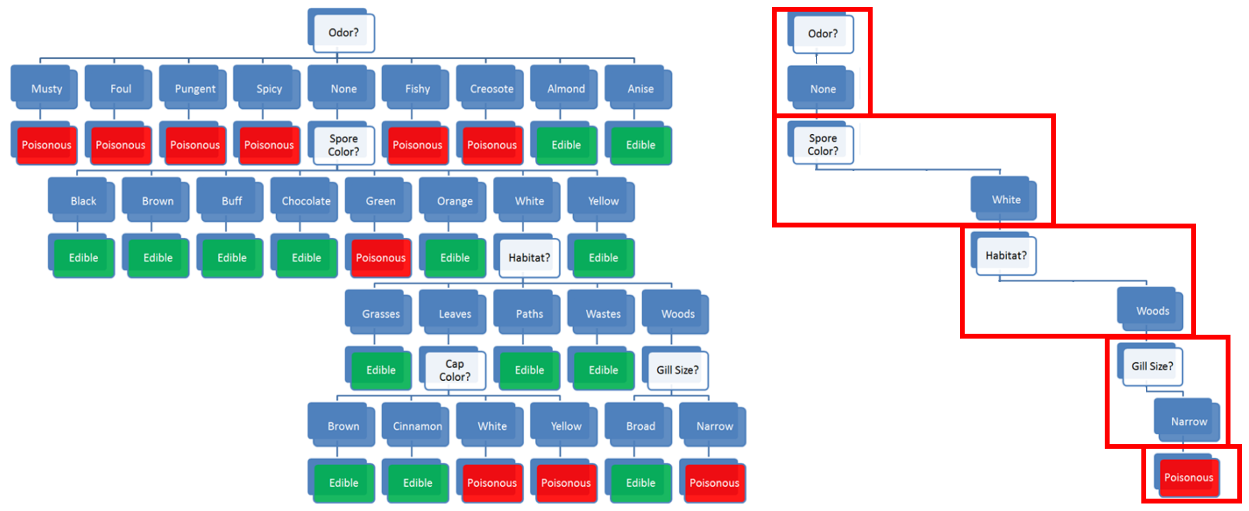


Figure 2. Decision tree for the mushroom classification problem, with decision path for **Amanita muscaria**.

The model prediction for **Amanita muscaria** follows the **decision path** shown on the right in Figure 2:

1. some mushroom **odors** (musty, spicy, etc.) are associated with poisonous mushrooms, some (almond, anise) with edible mushrooms, but there are mushrooms with no specific odor in either category – for mushroom with 'no odor' (as it he case with **Amanita muscaria**), odor does not provide enough information for proper classification and we need to incorporate additional features into the decision path;
2. among mushrooms with no specific odor, some **spore colours** (black, etc.) are associated with edible mushrooms, some (almond, anise) with poisonous mushrooms, but there are mushrooms with 'white' spores in either category – the combination 'no odor and white spores' does not provide enough information to classify **Amanita muscaria** and we need to incorporate additional features into the decision path;
3. among mushrooms of no specific odor with white spores, some **habitats** (grasses, paths, wastes) are associated with edible mushrooms, but there are mushrooms in either category that are found in the 'woods' – the combination 'no odor, white spores, found in the woods' does not provide enough information to classify **Amanita muscaria** and we need to incorporate additional features into the decision path;
4. among white-spored forest mushroom with no specific odor, a broad **gill size** is associated with edible mushrooms, whereas a 'narrow' gill size is associated with poisonous mushrooms – as **Amanita muscaria** is a narrow-gilled, white-spored forest mushroom with no specific odor, the decision path predicts that it is **poisonous**.

Note that the **cap color** does not affect the decision path.[3]

[3]It would have had **Amanita muscaria**'s habitat been 'leaves', however.

The model **does not explain why** this particular combinations of features is associated with poisonous mushrooms – the decision path is not **causal**.

At this point, a number of questions naturally arise:

- Would you have trusted an **edible** prediction?
- How are the features measured?
- What is the true cost of making a mistake?
- Is the data on which the model is built representative?
- What data is required to build trustworthy models?
- What do we need to know about the model in order to **trust it**?

3. Association Rules Mining

Tufte's Rejoinder

Correlation isn't causation. But it's a big hint.

– E. Tufte

Association rules discovery is a type of unsupervised learning that finds **connections** among the attributes and levels (and combinations thereof) of a dataset's observations.

For instance, we might analyse a (hypothetical) dataset on the physical activities and purchasing habits of North Americans and discover that

- runners who are also triathletes (the **premise**) tend to drive Subarus, drink microbrews, and use smart phones (the **conclusion**), or
- individuals who have purchased home gym equipment are unlikely to be using it 1 year later, say.

But the presence of a **correlation** between the premise and the conclusion does not necessarily imply the existence of a **causal relationship** between them.

It is rather difficult to "demonstrate" causation *via* data analysis; in practice, decision-makers pragmatically (and often erroneously) focus on the second half of Tufte's rejoinder, which asserts that "there's no smoke without fire."

Case in point, while being a triathlete does not cause one to drive a Subaru, Subaru Canada thinks that the connection is strong enough to offer to reimburse the registration fee at an IRONMAN 70.3 competition (since at least 2018)! [6]

Market Basket Analysis Association rules discovery is also known as **market basket analysis** after its original application, in which supermarkets record the contents of shopping carts (the **baskets**) at check-outs to determine which items are frequently purchased together.

For instance, while bread and milk might often be purchased together, that is unlikely to be of interest to supermarkets given the frequency of market baskets containing milk **or** bread (in the mathematical sense of "or").

Knowing that a customer has purchased bread does provide some information regarding whether they also purchased milk, but the individual probability that each item is found, separately, in the basket is so high to begin with that this insight is unlikely to be useful.

If 70% of baskets contain milk and 90% contain bread, say, we would expect **at least**

$$90\% \times 70\% = 63\%$$

of all baskets to contain milk **and** bread, should the presence of one in the basket be **totally independent** of the presence of the other.

If we then observe that 72% of baskets contain both items (a 1.15-fold increase on the expected proportion, assuming there is no link), we would conclude that there was at best a **weak correlation** between the purchase of milk and the purchase of bread.

Sausages and hot dog buns, on the other hand, which we might suspect are not purchased as frequently as milk and bread, might still be purchased as a pair more often than one would expect given the frequency of baskets containing sausages **or** buns.

If 10% of baskets contain sausages, and 5% contain buns, say, we would expect that

$$10\% \times 5\% = 0.5\%$$

of all baskets would contain sausages **and** buns, should the presence of one in the basket be **totally independent** of the presence of the other.

If we then observe that 4% of baskets contain both items (an 8-fold increase on the expected proportion, assuming there is no link), we would obviously conclude that there is a **strong correlation** between the purchase of sausages and the purchase of hot dog buns.

It is not too difficult to see how this information could potentially be used to help supermarkets turn a profit: announcing or advertising a sale on sausages while **simultaneously** (and quietly) raising the price of buns could have the effect of bringing in a higher number of customers into the store, increasing the sale volume for both items while keeping the combined price of the two items constant.[4]

A (possibly) apocryphal story shows the limitations of association rules: a supermarket found an association rule linking the purchase of beer and diapers and consequently moved its beer display closer to its diapers display, having confused correlation and causation.

Purchasing diapers does not cause one to purchase beer (or *vice-versa*); it could simply be that parents of newborns have little time to visit public houses and bars, and whatever drinking they do will be done at home. Who knows? Whatever the case, rumour has it that the experiment was neither popular nor successful.

Applications Typical uses include:

- finding **related concepts** in text documents – looking for pairs (triplets, etc) of words that represent a joint concept: {San Jose, Sharks}, {Michelle, Obama}, etc.;
- detecting **plagiarism** – looking for specific sentences that appear in multiple documents, or for documents that share specific sentences;
- identifying **biomarkers –** searching for diseases that are frequently associated with a set of biomarkers;
- making predictions and decisions based on association rules (there are pitfalls here);
- altering circumstances or environment to take advantage of these correlations (suspected causal effect);
- using connections to modify the likelihood of certain outcomes (see immediately above);
- imputing missing data,
- text autofill and autocorrect, etc.

Other uses and examples can be found in [5, 17, 48].

3.1 Causation and Correlation

Association rules can automate **hypothesis discovery**, but one must remain correlation-savvy (which is less prevalent among quantitative specialists than one might hope, in our experience). If attributes A and B are shown to be correlated in a dataset, there are four possibilities:

- A and B are correlated entirely by chance in this particular dataset;
- A is a relabeling of B (or *vice-versa*);
- A causes B (or *vice-versa*), or
- some combination of attributes $C_1, \ldots, C_n$ (which may not be available in the dataset) cause both A and B.

[4]The marketing team is banking on the fact that customers are unlikely to shop around to get the best deal on hot dogs AND buns, which may or may not be a valid assumption.

Siegel [48] illustrates the confusion that can arise with a number of real-life examples:

- Walmart has found that sales of strawberry Pop-Tarts increase about seven-fold in the days preceding the arrival of a hurricane;
- Xerox employees engaged in front-line service and sales-based positions who use Chrome and Firefox browsers perform better on employment assessment metrics and tend to stay with the company longer, or
- University of Cambridge researchers found that liking "Curly Fries" on Facebook is predictive of high intelligence.

It can be tempting to try to **explain** these results (again from [48]): perhaps

- when faced with a coming disaster, people stock up on comfort or nonperishable foods;
- the fact that an employee takes the time to install another browser shows that they are an informed individual and that they care about their productivity, or
- an intelligent person liked this Facebook page first, and her friends saw it, and liked it too, and since intelligent people have intelligent friends (?), the likes spread among people who are intelligent.

While these explanations *might* very well be the right ones (although probably not in the last case), there is **nothing in the data** that supports them. Association rules discovery **finds** interesting rules, but it does not explain them.

The point cannot be over-emphasized: correlation does not imply causation. Consultants and analysts might not have much control over the matter, but they should do whatever is in their power so that the following headlines do not see the light of day:

- "Pop-Tarts" get hurricane victims back on their feet;
- Using Chrome of Firefox improves employee performance, or
- Eating curly fries makes you more intelligent.

3.2 Definitions

A rule $X \to Y$ is a statement of the form "if X (the **premise**) then Y (the conclusion)" built from any logical combinations of a dataset attributes.

In practice, a rule **does not need to be true for all observations** in the dataset – there could be instances where the premise is satisfied but the conclusion is not.

In fact, some of the "best" rules are those which are only accurate 10% of the time, as opposed to rules which are only accurate is only 5% of the time, say. As always, **it depends on the context**.

To determine a rule's strength, we compute various rule metrics, such as the:

- **support** (coverage), which measures the frequency at which a rule occurs in a dataset – low coverage values indicate rules that rarely occur;
- **confidence** (accuracy), which measures the reliability of the rule: how often does the conclusion occur in the data given that the premises have occurred – rules with high confidence are "truer", in some sense;
- **interest**, which measures the difference between its confidence and the relative frequency of its conclusion – rules with high absolute interest are ... more interesting than rules with small absolute interest;
- **lift**, which measures the increase in the frequency of the conclusion which can be explained by the premises – in a rule with a high lift (> 1), the conclusion occurs more frequently than it would if it was independent of the premises;
- **conviction** [52], **all-confidence** [36], **leverage** [40], **collective strength** [3], and many others [19, 49].

In a dataset with N observations, let $\text{Freq}(A) \in \{0, 1, \ldots, N\}$ represent the count of the dataset's observations for which property A holds. This is all the information that is required to compute a rule's evaluation metrics:

$$\text{Support}(X \to Y) = \frac{\text{Freq}(X \cap Y)}{N} \in [0, 1]$$

$$\text{Confidence}(X \to Y) = \frac{\text{Freq}(X \cap Y)}{\text{Freq}(X)} \in [0, 1]$$

$$\text{Interest}(X \to Y) = \text{Confidence}(X \to Y) - \frac{\text{Freq}(Y)}{N} \in [-1, 1]$$

$$\text{Lift}(X \to Y) = \frac{N^2 \cdot \text{Support}(X \to Y)}{\text{Freq}(X) \cdot \text{Freq}(Y)} \in (0, N^2)$$

$$\text{Conviction}(X \to Y) = \frac{1 - \text{Freq(Y)}/N}{1 - \text{Confidence}(X \to Y)} \geq 0$$

Music Dataset A simple example will serve to illustrate these concepts. Consider a (hypothetical) music dataset containing data for $N = 15,356$ Chinese music lovers and a **candidate rule** RM:

> "If an individual is born before 1986 (X), then they own a copy of Teresa Teng's *The Moon Represents My Heart*, in some format (Y)".

Let's assume further that

- $\text{Freq}(X) = 3888$ individuals were born before 1986;
- $\text{Freq}(Y) = 9092$ individuals own a copy of *The Moon Represents My Heart*, and
- $\text{Freq}(X \cap Y) = 2720$ individuals were born before 1986 and own a copy of *The Moon Represents My Heart*.

We can easily compute the 5 metrics for RM:

$$\begin{aligned}
\text{Support(RM)} &= \frac{2720}{15,536} \approx 18\% \\
\text{Confidence(RM)} &= \frac{2720}{3888} \approx 70\% \\
\text{Interest(RM)} &= \frac{2720}{3888} - \frac{9092}{15,356} \approx 0.11 \\
\text{Lift(RM)} &= \frac{15,356^2 \cdot 0.18}{3888 \cdot 9092} \approx 1.2 \\
\text{Conviction(RM)} &= \frac{1-9092/15,356}{1-2720/3888} \approx 1.36
\end{aligned}$$

These values are easy to interpret: RM occurs in **18%** of the dataset's instances, and it holds true in **70%** of the instances where the individual was born prior to 1986.

This would seem to make RM a **meaningful rule** about the dataset – being older and owning that song are linked properties. But if being younger and not owning that song are not also linked properties, the statement is actually weaker than it would appear at a first glance.

As it happens, RM's lift is **1.2**, which can be rewritten as

$$1.2 \approx \frac{0.70}{0.56},$$

i.e. 56% of younger individuals also own the song.

The ownership rates between the two age categories are different, but perhaps not as significantly as one would deduce using the confidence and support alone, which is reflected by the rule's "low" interest, whose value is **0.11**.

Finally, the rule's conviction is **1.36**, which means that the rule would be incorrect 36% more often if X and Y were completely independent.

All this seems to point to the rule RM being not entirely devoid of meaning, but to what extent, exactly? **This is a difficult question to answer.**[5]

It is nearly impossible to provide **hard** and **fast** thresholds: it always depends on the context, and on comparing evaluation metric values for a rule with the values obtained for some other of the dataset's rules. In short, evaluation of a lone rule is **meaningless**.

3.3 Generating Rules

Given association rules, it is straightforward to evaluate them using various metrics, as discussed in Section 3.2.

The real challenge of association rules discovery lies in **generating** a set of candidate rules which are likely to be retained, without wasting time generating rules which are likely to be discarded.

[5]There will be times when an interest of 0.11 in a rule would be considered a smashing success; a lift of 15 would not be considered that significant but a support of 2% would be, and so forth.

An **itemset** (or instance set) for a dataset is a list of attributes and values. A set of **rules** can be created from the itemset by adding 'IF ... THEN' blocks to the instances.

As an example, from the instance set

$$\begin{aligned}\{&\text{membership} = \text{True},\\ &\text{age} = \text{Youth},\\ &\text{purchasing} = \text{Typical}\},\end{aligned}$$

we can create the 7 following 3—item rules:

- IF (membership = True AND age = Youth) THEN purchasing = Typical;
- IF (age = Youth AND purchasing = Typical) THEN membership = True;
- IF (purchasing = Typical AND membership = True) THEN age = Youth;
- IF membership = True THEN (age = Youth AND purchasing = Typical);
- IF age = Youth THEN (purchasing = Typical AND membership = True);
- IF purchasing = Typical THEN (membership = True) AND age = Youth);
- IF ∅ THEN (membership = True AND age = Youth AND purchasing = Typical);

the 6 following 2—item rules:

- IF membership = True THEN purchasing = Typical;
- IF age = Youth THEN membership = True;
- IF purchasing = Typical THEN age = Youth;
- IF ∅ THEN (age = Youth AND purchasing = Typical);
- IF ∅ THEN (purchasing = Typical AND membership = True);
- IF ∅ THEN (membership = True) AND age = Youth);

and the 3 following 1—item rules:

- IF ∅ THEN age = Youth;
- IF ∅ THEN purchasing = Typical;
- IF ∅ THEN membership = True.

In practice, we usually only consider rules with the same number of items as there are members in the itemset: in the example above, for instance, the 2—item rules could be interpreted as emerging from the 3 separate itemsets

$$\begin{aligned}&\{\text{membership} = \text{True}, \text{age} = \text{Youth}\}\\ &\{\text{age} = \text{Youth}, \text{purchasing} = \text{Typical}\}\\ &\{\text{purchasing} = \text{Typical}, \text{membership} = \text{True}\},\end{aligned}$$

and the 1—item rules as arising from the 3 separate itemsets

$$\begin{aligned}&\{\text{membership} = \text{True}\}\\ &\{\text{age} = \text{Youth}\}\\ &\{\text{purchasing} = \text{Typical}\}.\end{aligned}$$

Note that rules of the form $\varnothing \to X$ (or IF $\varnothing$ THEN X) are typically denoted simply by X.

Now, consider an itemset $\mathscr{C}_n$ with n members (that is to say, n attribute/level pairs).

In an n—item rule derived from $\mathscr{C}$, each of the n members appears either in the premise or in the conclusion; there are thus 2^n such rules, in principle.

The rule where each member is part of the premise (i.e., the rule without a conclusion) is nonsensical and is not allowed; we can derive exactly $2^n - 1$ n—item rules from $\mathscr{C}_n$.

Thus, the **number of rules increases exponentially** when the **number of features increases linearly**.

This combinatorial explosion is a problem – it instantly disqualifies the **brute force** approach (simply listing all possible itemsets in the data and generating all rules from those itemsets) for any dataset with a realistic number of attributes.

How can we then generate a small number of **promising** candidate rules, in general?

The ***apriori*** algorithm is an early attempt to overcome that difficulty. Initially, it was developed to work for **transaction data** (i.e. goods as columns, customer purchases as rows), but every reasonable dataset can be transformed into a transaction dataset using dummy variables.

The algorithm attempts to find **frequent itemsets** from which to build candidate rules, instead of building rules from **all** possible itemsets.

It starts by identifying frequent **individual items** in the database and extends those that are retained into larger and larger **item supersets**, who are themselves retained only if they occur **frequently enough** in the data.

The main idea is that "all non-empty subsets of a frequent itemset must also be frequent" [9], or equivalently, that all supersets of an infrequent itemset must also be infrequent (see Figure 3).

In the technical jargon of machine learning, we say that *apriori* uses a **bottom-up approach** and the **downward closure property of support**.

The memory savings arise from the fact that the algorithm prunes candidates with **infrequent sub-patterns** and removes them from consideration for any future itemset: if a 1—itemset is not considered to be frequent enough, any 2—itemset containing it is also infrequent (see Table 1 for another illustration).

Of course, this requires a support threshold **input**, for which there there is no guaranteed way to pick a "good" value; it has to be set sufficiently high to minimize the number of frequent itemsets that are being considered, but not so high that it removes too many candidates from the **output list**; as ever, optimal threshold values are **dataset-specific**.

The algorithm terminates when no further itemsets extensions are retained, which always occurs given the finite number of levels in categorical datasets (see below).

- **Strengths:** easy to implement and to parallelize [32];
- **Limitations:** slow, requires frequent data set scans, not ideal for finding rules for infrequent and rare itemsets.

More efficient algorithms have since displaced it in practice, although it retains historical value:

- **Max-Miner** tries to identify frequent itemsets without enumerating them – it performs jumps in space instead of using bottom-up approach;
- **Eclat** is faster and uses depth-first search, but requires extensive memory storage (apriori and eclat are both implemented in the `R` package `arules`).

Notes How **reliable** are association rules? What is the likelihood that they occur entirely **by chance**? How **relevant** are they? Can they be generalised **outside** the dataset, or to **new** data streaming in?

These questions are notoriously difficult to solve for association rules discovery, but **statistically sound association discovery** can help reduce the risk of finding spurious associations to a user-specified significance level [19, 49].

We end this section with a few comments:

- Since frequent rules correspond to instances that occur repeatedly in the dataset, algorithms that generate itemsets often try to **maximise coverage**. When **rare events** are more meaningful (such as detection of a rare disease or a threat), we need algorithms that can generate rare itemsets. **This is not a trivial problem**.
- Continuous data has to be binned into **categorical** data to generate rules. As there are many ways to accomplish that task, the same dataset can give rise to completely different rules. This could create some credibility issues with the client.
- Other popular algorithms include: AIS, SETM, aprioriTid, aprioriHybrid, PCY, Multistage, Multihash, etc.
- Additional evaluation metrics can be found in the `arules` documentation [36].

3.4 Toy Example: Titanic Dataset

Compiled by Robert Dawson in 1995, the *Titanic* dataset consists of 4 categorical attributes for each of the 2201 people aboard the Titanic when it sank in 1912 (some issues with the dataset have been documented, but we will ignore them for now):

- **class** (1st class, 2nd class, 3rd class, crewmember)
- **age** (adult, child)
- **sex** (male, female)
- **survival** (yes, no)

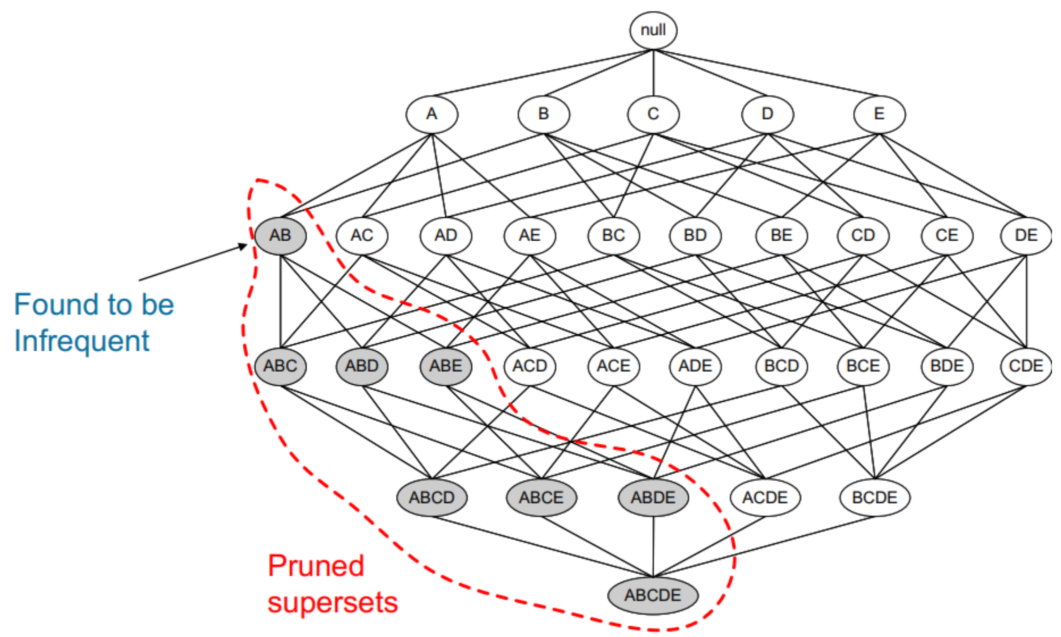
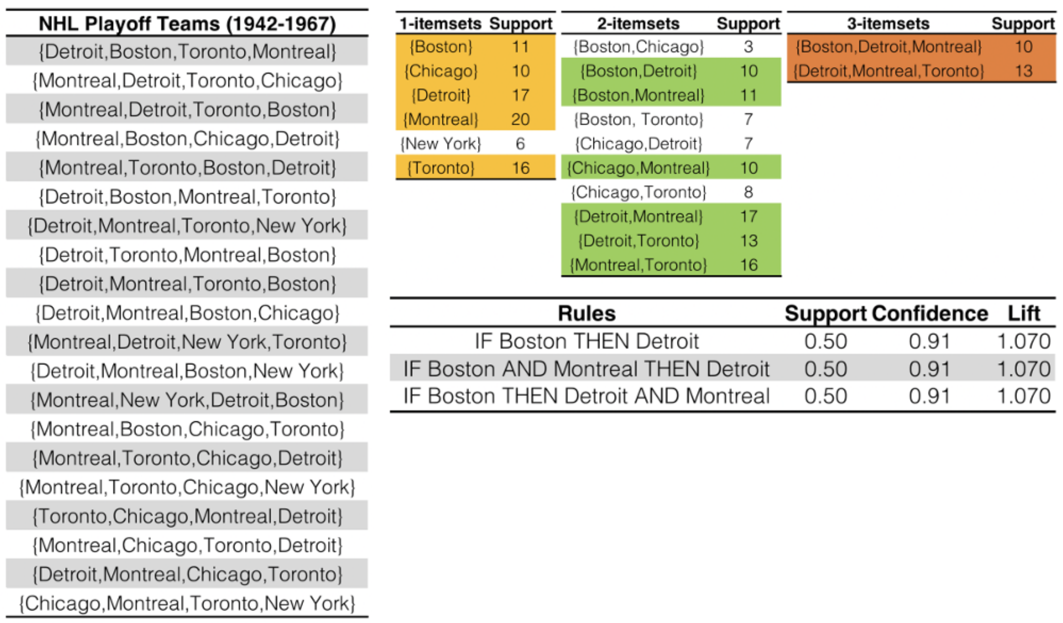


Figure 3. Pruned supersets of an infrequent itemset in the apriori network of a dataset with 5 items [9]; no rule would be generated from the grey itemsets.



NHL Playoff Teams (1942-1967)
{Detroit,Boston,Toronto,Montreal}
{Montreal,Detroit,Toronto,Chicago}
{Montreal,Detroit,Toronto,Boston}
{Montreal,Boston,Chicago,Detroit}
{Montreal,Toronto,Boston,Detroit}
{Detroit,Boston,Montreal,Toronto}
{Detroit,Montreal,Toronto,New York}
{Detroit,Toronto,Montreal,Boston}
{Detroit,Montreal,Toronto,Boston}
{Detroit,Montreal,Boston,Chicago}
{Montreal,Detroit,New York,Toronto}
{Detroit,Montreal,Boston,New York}
{Montreal,New York,Detroit,Boston}
{Montreal,Boston,Chicago,Toronto}
{Montreal,Toronto,Chicago,Detroit}
{Montreal,Toronto,Chicago,New York}
{Toronto,Chicago,Montreal,Detroit}
{Montreal,Chicago,Toronto,Detroit}
{Detroit,Montreal,Chicago,Toronto}
{Chicago,Montreal,Toronto,New York}

1-itemsets	Support
{Boston}	11
{Chicago}	10
{Detroit}	17
{Montreal}	20
{New York}	6
{Toronto}	16

2-itemsets	Support
{Boston,Chicago}	3
{Boston,Detroit}	10
{Boston,Montreal}	11
{Boston, Toronto}	7
{Chicago,Detroit}	7
{Chicago,Montreal}	10
{Chicago,Toronto}	8
{Detroit,Montreal}	17
{Detroit,Toronto}	13
{Montreal,Toronto}	16

3-itemsets	Support
{Boston,Detroit,Montreal}	10
{Detroit,Montreal,Toronto}	13

Rules	Support	Confidence	Lift
IF Boston THEN Detroit	0.50	0.91	1.070
IF Boston AND Montreal THEN Detroit	0.50	0.91	1.070
IF Boston THEN Detroit AND Montreal	0.50	0.91	1.070

Table 1. Association rules for NHL playoff teams (1942-1967). A list of the 4 teams making the playoffs each year is shown on the left ($N = 20$). Frequent itemsets are generated using the *apriori* algorithms, with a support threshold of 10. We see that there are 5 frequent 1—itemsets, top row, in yellow (New York made the playoffs $6 < 10$ times – no larger frequent itemset can contain New York). 6 frequent 2—itemsets are found in the subsequent list of ten 2—itemsets, top row, in green (note the absence of New York). Only 2 frequent 3—itemsets are found, top row, in orange. Candidate rules are generated from the shaded itemsets; the rules retained by the thresholds Support ≥ 0.5, Confidence ≥ 0.7, and Lift > 1 (barely) are shown in the table on the bottom row – the main result is that when Boston made the playoffs, it was not surprising to see Detroit also make the playoffs (the presence or absence of Montreal in a rule is a red herring, as Montreal made the playoffs every year in the data. Are these rules meaningful?

Rule	Supp	Conf	Lift
IF class = 2nd AND age = Child THEN survived = Yes	0.01	1	3.10
IF class = 1st AND sex = Female THEN survived = Yes	0.06	0.97	3.01
IF class = 2nd AND sex = Female THEN survived = Yes	0.04	0.88	2.72
IF class = Crew AND sex = Female THEN survived = Yes	0.00	0.87	2.70
IF class = 2nd AND sex = Male AND age = Adult THEN survived = No	0.07	0.92	1.35
IF class = 2nd AND sex = Male THEN survived = No	0.07	0.86	1.27
IF class = 3rd AND sex = Male AND age = Adult THEN survived = No	0.18	0.84	1.24
IF class = 3rd AND sex = Male THEN survived = No	0.19	0.83	1.22

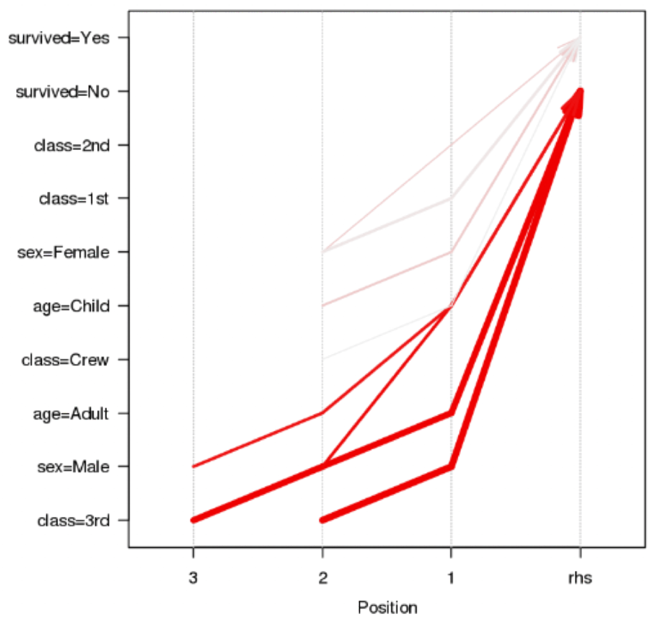


Figure 4. Visualisations of the 8 *Titanic* association rules with parallel coordinates.

The natural question of interest for this dataset is

"how does survival relate to the other attributes?"

This is not, strictly speaking, an unsupervised task (as the interesting rules' structure is fixed to conclusions of the form survival = Yes or survival = No).

For the purpose of this example, we elect not to treat the problem as a **predictive task**, since the situation on the Titanic has little bearing on survival for new data – as such, we use fixed-structure association rules to **describe** and **explore** survival conditions on the *Titanic* (compare with [44]).

We use the `arules` implementation of the *apriori* algorithm in `R` to generate and prune candidate rules, eventually leading to **8 rules** (the results can be visualised in Figure 4). Who survived?[6]

3.5 Case Study: Danish Medical Data

In *temporal disease trajectories condensed from population wide registry data covering 6.2 million patients* [26], A.B. Jensen et al. study diagnoses in the Danish population, with the help of association rules mining and clustering methods (see Section 5).

Objectives Estimating **disease progression** (trajectories) from current patient state is a crucial notion in medical studies. Such trajectories had (at the time of publication) only been analyzed for a small number of diseases, or using large-scale approaches without consideration for time exceeding a few years.

Using data from the *Danish National Patient Registry* (an extensive, long-term data collection effort by Denmark), the authors sought connections between different **diagnoses**: how does the presence of a diagnosis at some point in time allow for the prediction of another diagnosis at a later point in time?

Methodology The authors took the following methodological steps:

1. compute the **strength of correlation** for pairs of diagnoses over a 5 year interval (on a representative subset of the data);
2. test diagnoses pairs for **directionality** (one diagnosis repeatedly occurring before the other);
3. determine reasonable **diagnosis trajectories** (thoroughfares) by combining smaller (but frequent) trajectories with overlapping diagnoses;
4. **validate** the trajectories by comparison with non-Danish data;
5. **cluster** the thoroughfares to identify a small number of **central medical conditions** (key diagnoses) around which disease progression is organized.

Data The Danish National Patient Registry is an electronic health registry containing administrative information and diagnoses, covering the whole population of Denmark, including private and public hospital visits of all types: inpatient (overnight stay), outpatient (no overnight stay) and emergency. The data set covers 15 years, from January '96 to November '10 and consists of 68 million records for 6.2 million patients.

[6]For the last time, **correlation does not imply causation.**

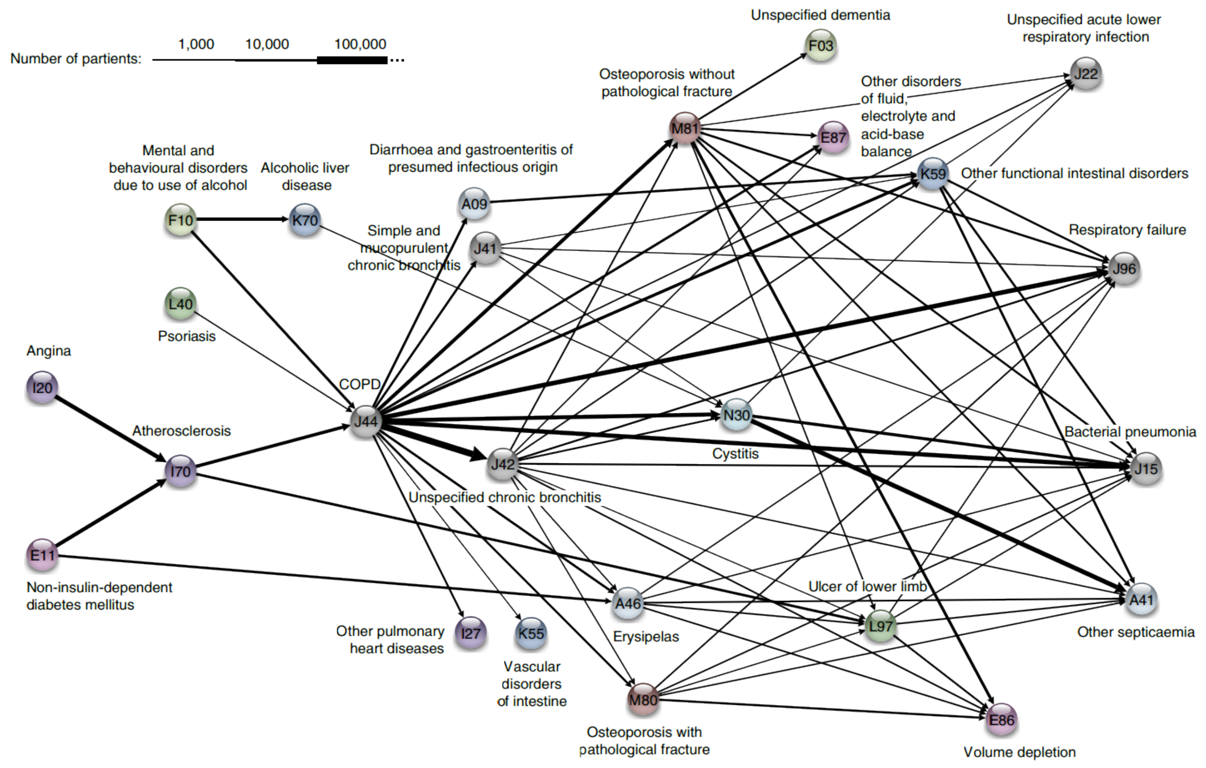

Number of partients:
1,000
10,000
100,000
Unspecified dementia
F03
Unspecified acute lower respiratory infection
J22
Osteoporosis without pathological fracture
M81
Other disorders of fluid, electrolyte and acid-base balance
E87
K59
Other functional intestinal disorders
Mental and behavioural disorders due to use of alcohol
F10
Alcoholic liver disease
K70
Diarrhoea and gastroenteritis of presumed infectious origin
A09
Simple and mucopurulent chronic bronchitis
J41
Respiratory failure
J96
L40
Psoriasis
Angina
I20
COPD
J44
Atherosclerosis
I70
N30
Cystitis
Bacterial pneumonia
J15
J42
Unspecified chronic bronchitis
E11
Non-insulin-dependent diabetes mellitus
Ulcer of lower limb
A46
A41
L97
Other pulmonary heart diseases
I27
K55
Erysipelas
Other septicaemia
Vascular disorders of intestine
M80
E86
Osteoporosis with pathological fracture
Volume depletion

Figure 5. The COPD cluster showing five preceding diagnoses leading to COPD and some of the possible outcomes [26].

Challenges and Pitfalls

- Access to the **Patient Registry** is protected and could only be granted after approval by the *Danish Data Registration Agency the National Board of Health*.
- Gender-specific differences in diagnostic trends are clearly identifiable (pregnancy and testicular cancer do not have much cross-appeal), but many diagnoses were found to be made exclusively (or at least, predominantly) in different sites (inpatient, outpatient, emergency ward), which suggests the importance of stratifying by **site** as well as by **gender**.
- In the process of forming small diagnoses chains, it became necessary to compute the correlations using **large groups** for each pair of diagnoses. For close to 1 million diagnosis pairs, more than 80 million samples would have been required to obtain significant p—values while compensating for **multiple testing**, which would have translated to a few thousand years' worth of computer running time. A pre-filtering step was included to avoid this pitfall.[7]

Project Summaries and Results The dataset was reduced to **1,171 significant trajectories**. These thoroughfares were clustered into patterns centred on 5 key diagnoses central to disease progression: **diabetes**, **chronic obstructive pulmonary disease** (COPD), **cancer**, **arthritis**, and **cerebrovascular disease**.

[7]The final trajectories were validated using the full sampling procedure.

Early diagnoses for these central factors can help reduce the risk of adverse outcome linked to future diagnoses of other conditions. Two author quotes illustrate the importance of these results:

> "The sooner a health risk pattern is identified, the better we can prevent and treat critical diseases."
> — S. Brunak

> "Instead of looking at each disease in isolation, you can talk about a complex system with many different interacting factors. By looking at the order in which different diseases appear, you can start to draw patterns and see complex correlations outlining the direction for each individual person."
> — L.J. Jensen

Among the specific results, the following "surprising" insights were found:

- a diagnosis of anemia is typically followed months later by the discovery of colon cancer;
- gout was identified as a step on the path toward cardiovascular disease, and
- COPD is under-diagnosed and under-treated.

The disease trajectories cluster for COPD is shown in Figure 5.

4. Supervised Learning and Classification

Embarrassment of Riches

"The diversity of problems that can be addressed by classification algorithms is significant, and covers many domains. It is difficult to comprehensively discuss all the methods in a single book."

– C.C. Aggarwal [1]

The principles underlying classification, regression and class probability estimation are well-known and straightforward.

Classification is a supervised learning endeavour in which a sample set of data (the **training set**) is used to determine rules and patterns that divide the data into predetermined groups, or classes. The training data usually consists of a **randomly** selected subset of the **labeled** data.[8]

In the **testing phase**, the model is used to assign a class to observations in the **testing set**, in which the label is hidden, in spite of being actually known.

The performance of the predictive model is then **evaluated** by comparing the predicted and the values for the testing set observations (but **never** using the training set observations).

A number of technical issues need to be addressed in order to achieve optimal performance, among them: determining which features to select for inclusion in the model and, perhaps more importantly, which algorithm to choose.

The mushroom (classification) model from Section 2 provides a clean example of a **classification model**, albeit one for which no detail regarding the training data and choice of algorithm were made available.

Applications Classification and value estimation models are among the most frequently used of the data science models, and form the backbone of what is also known as **predictive analytics**. There are applications and uses in just about every field of human endeavour, such as:

- **medicine and health science** – predicting which patient is at risk of suffering a second, and this time fatal, heart attack within 30 days based on health factors (blood pressure, age, sinus problems, etc.);
- **social policies** – predicting the likelihood of required assisting housing in old age based on demographic information/survey answers;
- **marketing/business** – predicting which customers are likely to cancel their membership to a gym based on demographics and usage;
- in general, predicting that an object belongs to a particular class, or organizing and grouping instances into categories, or
- enhancing the detection of relevant objects:
  - **avoidance** – "this object is an incoming vehicle";
  - **pursuit** – "this borrower is unlikely to default on her mortgage";
  - **degree** – "this dog is 90% likely to live until it's 7 years old";
- **economics** – predicting the inflation rate for the coming two years based on a number of economic indicators.

[8] **Value estimation** (regression) is similar to classification, except that the target variable is numerical.

Other examples may be found in [15, 27–29].

Some concrete examples may provide a clearer picture of the types of supervised learning problems that quantitative consultants may be called upon to solve.

- A motor insurance company has a fraud investigation department that studies up to 20% of all claims made, yet money is still getting lost on fraudulent claims. To help better direct the investigators, management would like to determine, using past data, if it is possible to predict
  - whether a claim is likely to be fraudulent?
  - whether a customer is likely to commit fraud in the near future?
  - whether an application for a policy is likely to result in a fraudulent claim?
  - the amount by which a claim will be reduced if it is fraudulent?
- Customers who make a large number of calls to a mobile phone company's customer service number have been identified as churn risks. The company is interested in reducing said churn. Can they predict
  - the overall lifetime value of a customer?
  - which customers are more likely to churn in the near future?
  - what retention offer a particular customer will best respond to?

In every classification scenario, the following questions must be answered before embarking on analysis:

- What kind of data is **required**?
- How much of it?
- What would constitute a **predictive success**?
- What are the **risks** associated with a predictive modeling approach?

These have no one-size-fits-all answers.

In the absence of testing data, classification models cannot be used for predictive tasks, but may still be useful for **descriptive** tasks. When testing data exists, the process is often quite similar, independently of the choice of the algorithm (see the classification pipeline shown in Figure 6).

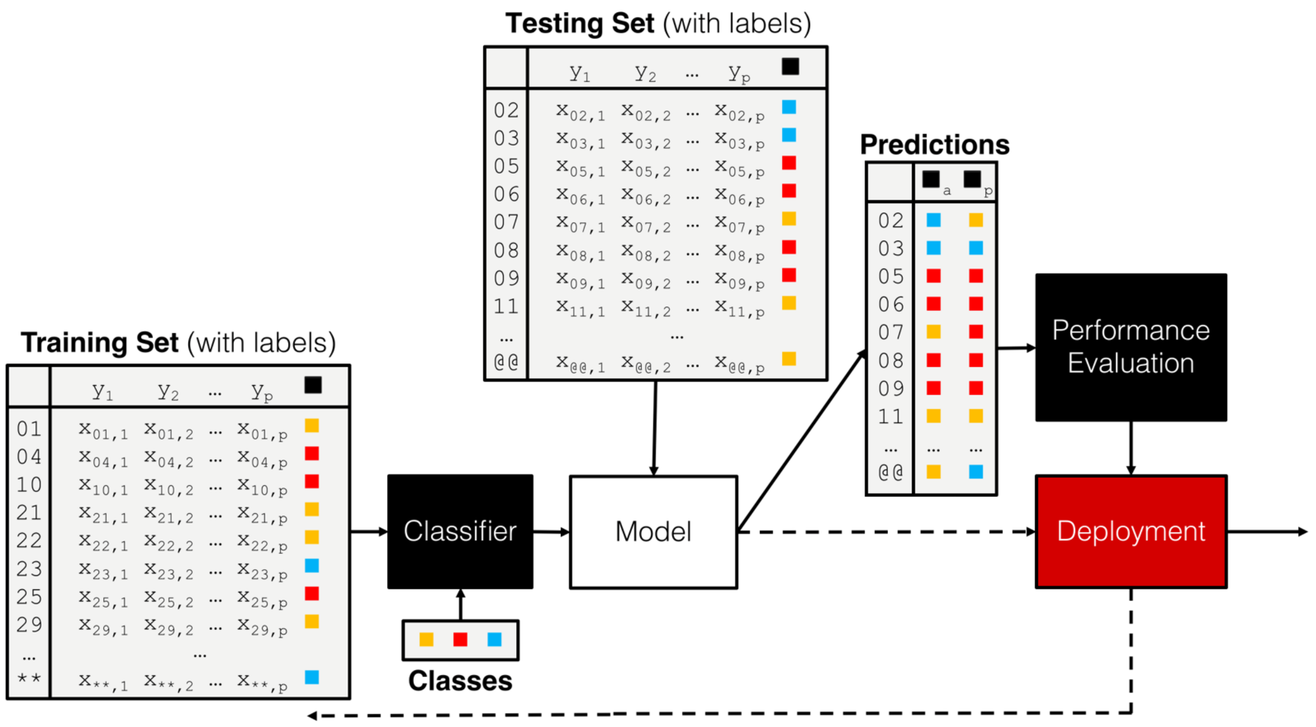


Figure 6. A classification pipeline, including training set, testing set, performance evaluation, and (eventual) deployment.

4.1 Classification Algorithms

The number of classification algorithms is truly staggering – it often seems as though new algorithms and variants are put forward on a monthly basis, depending on the task and on the type of data [1].

While some of them tend to be rather esoteric, there is a fairly small number of commonly-used workhorse algorithms/approaches that data scientists and consultants should at least have at their command (full descriptions are available in [21, 42, 50]):

- **logistic regression** and linear regression are classical models which are often used by statisticians but rarely in a classification setting (the estimated coefficients are often used to determine the features' importance); one of their strengths is that the machinery of standard statistical theory (hypothesis testing, confidence intervals, etc.) is still available to the analyst, but they are easily affected by variance inflation in the presence of predictor multi-colinearity, and the stepwise variable selection process that is typically used is problematic – regularization methods would be better suited [22] (see Figure 7, top left);
- **neural networks** have become popular recently due to the advent of deep learning; they might provide the prototypical example of a **black box** algorithm as they are hard to interpret; another issue is that they require a fair amount of data to train properly – we will have more to say on the topic in a later chapter;
- **decision trees** are perhaps the most commons of all data science algorithms, but they tend to overfit the data when they are not pruned correctly, a process which often has to be done manually (see Figure 7) – we shall discuss the pros an cons of decision trees in general in Section 4.2;
- **naïve Bayes classifiers** have known quite a lot of success in text mining applications (more specifically as the basis of powerful spam filters), but, embarrassingly, no one is quite sure why they should work as well as they do given that one of their required assumptions (independence of priors) is rarely met in practice (see Figure 7, top right);
- **support vector machines** attempt to separate the dataset by "fitting" as wide of a "tube" as possible through the classes (subjected to a number of penalty constraints); they have also known successes, most notably in the field of digital handwriting recognition, but their decision boundaries (the tubes in question) tend to be non-linear and quite difficult to interpret; nevertheless, they may help mitigate some of the difficulties associated with big data (see Figure 8);
- **nearest neighbours classifiers** basically implement a voting procedure and require very little assumptions about the data, but they are not very stable as adding training points may substantially modify the boundary (see Figure 7 and Figure 8, middle row).

Boosting methods [31] and **Bayesian methods** (see later chapter) are also becoming more popular.

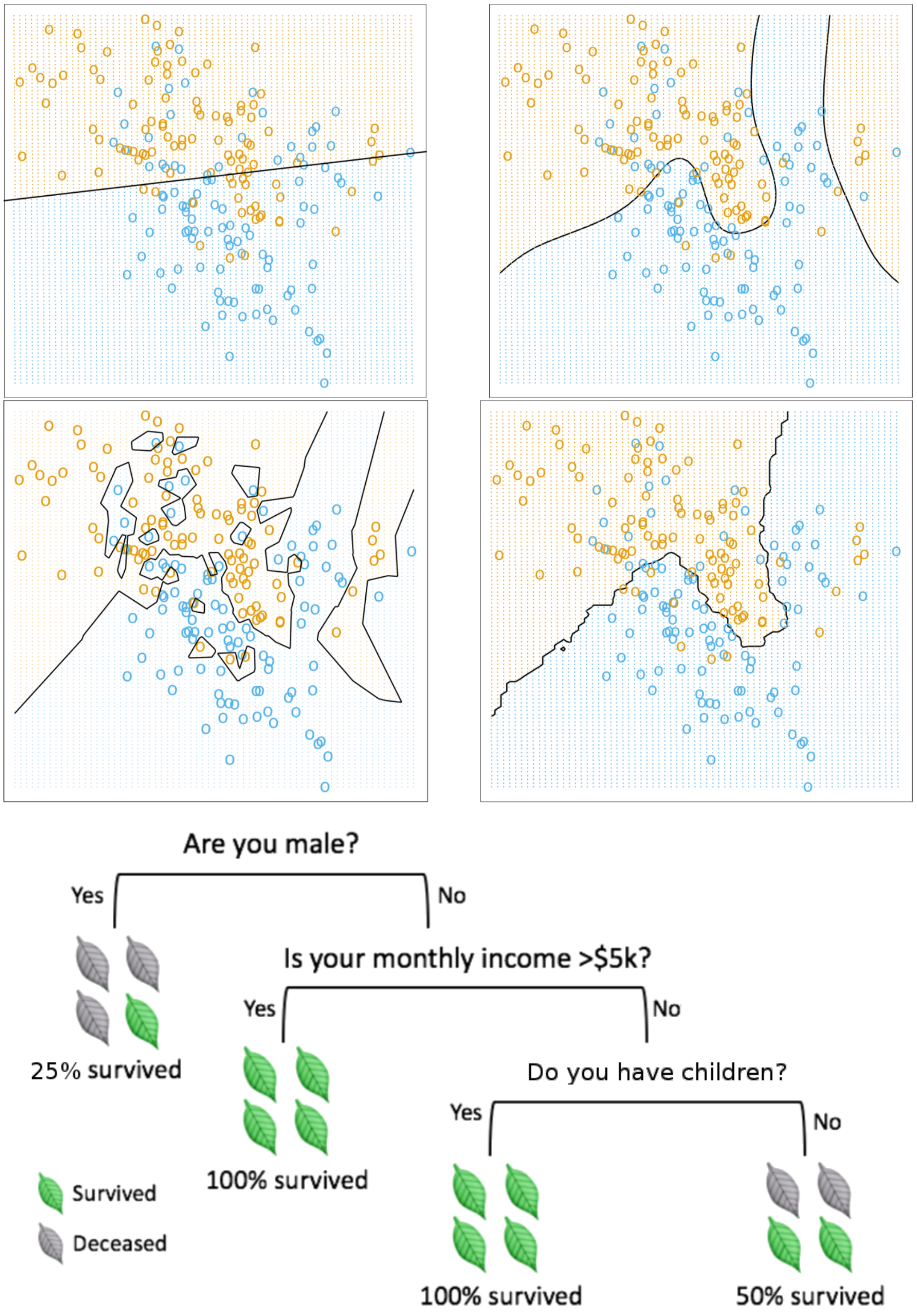


Figure 7. Illustrations of various classifiers (linear regression, top left; optimal Bayes, top right; 1NN and 15NN, middle left and right, respectively, on an artificial dataset (from [21]); decision tree depicting the chances of survival for various disasters (fictional, based on [35]). Note that linear regression is more stable, simpler to describe, but less accurate than *k*NN and optimal Bayes.

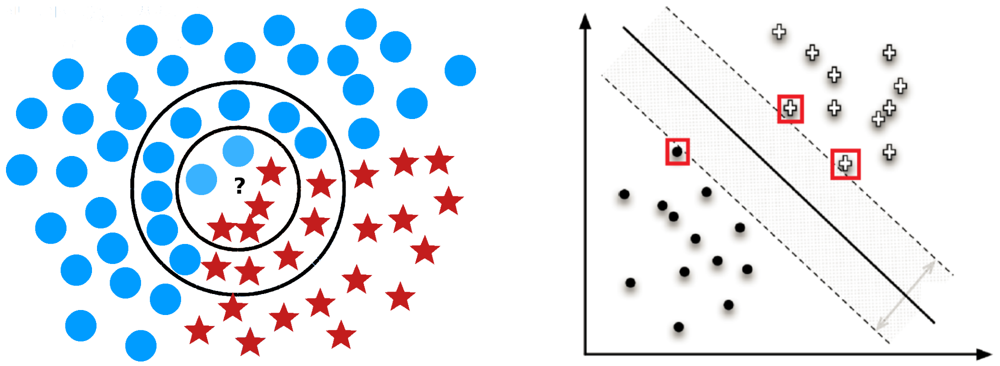


Figure 8. Illustration of a k nearest neighbour (left) and a support vector machines classifier (right, based on [42]). What is the 6NN prediction for the location marked by a question mark? What about the 19NN prediction?

4.2 Decision Trees

In order to highlight the relative simplicity of most classification algorithms, we will discuss the workings of **ID3**, a historically significant decision tree algorithm.[9]

Classification trees are perhaps the most **intuitive** of all supervised methods: classification is achieved by following a path up the tree, from its **root**, through its **branches**, and ending at its **leaves** (alghough typically the tree is depicted with its root at the top and its leaves at the bottom).

To make a **prediction** for a new instance, it suffices to follow the path down the tree, reading the prediction directly once a leaf is reached. It sounds simple enough in theory, but in practice, creating the tree and traversing it might be **time-consuming** if there are too many variables in the dataset (due to the criterion that is used to determine how the branches split).

Prediction accuracy can be a concern in trees whose growth is **unchecked**. In practice, the criterion of **purity** at the leaf-level (that is to say, all instances in a leaf belong to the same leaf) is linked to bad prediction rates for new instances. Other criteria are often used to prune trees, which may lead to impure leaves.

How do we grow such trees? For predictive purposes, we need a training set and a testing set upon which to evaluate the tree's performance. Ross Quinlan's **Iterative Dichotomizer 3** (a precursor to the widely-used C4.5 and C5.0) follows a simple procedure:

1. split the training data (**parent**) set into (**children**) subsets, using the different levels of a particular attribute;
2. compute the **information gain** for each subset;
3. select the **most advantageous** split, and
4. repeat for each node until some **leaf criterion** is met.

[9]ID3 would never be used in a deployment setting, but it will serve to illustrate a number of classification concepts.

Entropy is a measure of disorder in a set S. Let p_i be the proportion of observations in S belonging to category i, for $i = 1,\ldots,n$. The entropy of S is given by

$$E(S) = -\sum_{i=1}^{n} p_i \log p_i.$$

If the **parent set** S consisting of m records is split into k children sets $C_1,\ldots,C_k$ containing $q_1,\ldots,q_k$ records, respectively, then the **information gained** from the split is

$$I(S : C_1,\ldots,C_k) = E(S) - \frac{1}{m}\sum_{j=1}^{k} q_j E(C_j).$$

The sum term in the information gain equation is a weighted average of the entropy of the children sets. If the split leads to little disorder in the children, then $\text{IG}(S; C_1,\ldots,C_k)$ is high; if the split leads to similar disorder in both children and parent, then $\text{IG}(S; C_1,\ldots,C_k)$ is low.

Consider, as in Figure 9, two splits shown for a parent set with 30 observations separated into 2 classes: $\circ$ and $\star$.

Visually, it appears as though the binary split does a better job of separating the classes. Numerically, the entropy of the parent set S is

$$\begin{aligned} E(S) &= -p_\circ \log p_\circ - p_\star \log p_\star \\ &= -\frac{16}{30}\log\frac{16}{30} - \frac{14}{30}\log\frac{14}{30} \approx 0.99. \end{aligned}$$

For the binary split (on the left), leading to the children set L (left) and R (right), the respective entropies are

$$E(L) = -\frac{12}{13}\log\frac{12}{13} - \frac{1}{13}\log\frac{1}{13} \approx 0.39$$

and

$$E(R) = -\frac{4}{17}\log\frac{4}{17} - \frac{13}{17}\log\frac{13}{17} \approx 0.79,$$

so that the information gained by that split is

$$\text{IG}(S; C_L, C_R) \approx 0.99 - \frac{1}{30}[13 \cdot 0.39 + 17 \cdot 0.79] = 0.37.$$

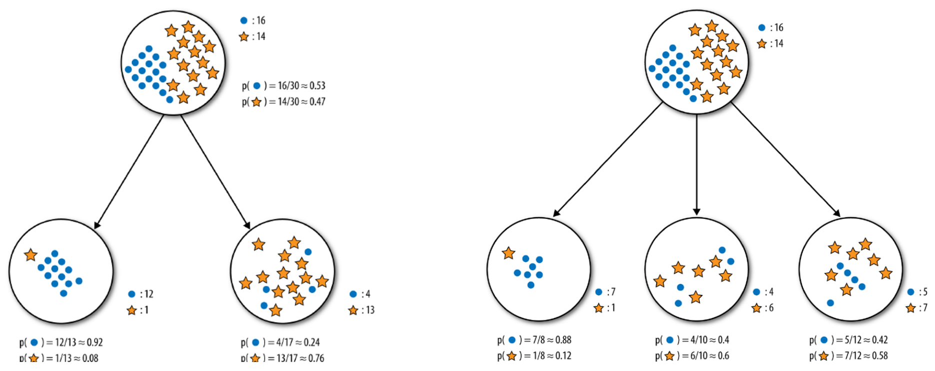


Figure 9. Picking the optimal information gain split (from [42]).

On its own, this value is not substantially meaningful – it is in comparison to the information gained from other splits that it becomes useful.

A similar computation for the ternary split leads to $\text{IG}(S; C_1, C_2, C_3) \approx 0.13$, which is indeed smaller than the information gained by the binary split – of these two options, ID3 would select the first as being **most advantageous**.

Decision trees have numerous strengths: they

- are easy to interpret, providing, as they do, a **white box model** – predictions can always be explained by following the appropriate paths;
- can handle numerical and categorical data **simultaneously**, without first having to “binarise” the data;
- can be used with **incomplete** datasets, if needed (although there is still some value in imputing missing observations);
- allow for **built-in feature selection** as less relevant features do not tend to be used as splitting features;
- make **no assumption** about independence of observations, underlying distributions, multi-colinearity, etc., and can thus be used without the need to verify assumptions;
- lend themselves to **statistical validation** (in the form of cross-validation), and
- are in line with **human decision-making approaches**, especially when such decisions are taken deliberately.

On the other hand, they are

- **not usually as accurate** as other more complex algorithms, nor **as robust**, as small changes in the training data can lead to a completely different tree, with a completely different set of predictions;[10]
- particularly **vulnerable to overfitting** in the absence of pruning – and pruning procedures are typically fairly convoluted (some algorithms automate this process, using statistical tests to determine when a tree’s “full” growth has been achieved), and
- **biased** towards categorical features with high number of levels, which may give such variables undue importance in the classification process.

Information gain tends to grow small trees in its puruit of pure leaves, but it is not the only splitting metric in use (**Gini impurity**, **variance reduction**, etc.).

Notes ID3 is a precursor of C4.5, perhaps the most popular decision tree algorithm on the market. There are other tree algorithms, such as C5.0, CHAID, MARS, conditional inference trees, CART, etc., each grown using algorithms with their own strengths and weaknesses.

Regression trees are grown in a similar fashion, but with a **numerical** response variable (predicted inflation rate, say), which introduces some complications [21, 24].

Decision trees can also be combined together using **boosting algorithms** (such as **AdaBoost**) or **random forests**, providing a type of voting procedure also known as **ensemble learning** – an individual tree might make middling predictions, but a large number of judiciously selected trees are likely to make good predictions, on average [21, 24, 31].

Additionally:

- since classification is linked to **probability estimation**, approaches that extend the basic ideas of regression models could prove fruitful;
- **rare occurrences** are often more interesting and more difficult to predict and identify than regular instances – historical data at Fukushima’s nuclear reactor prior to the 2011 meltdown could not have been used to learn about meltdowns, for obvious reasons;[11]
- with big datasets, algorithms must also consider **efficiency** – thankfully, decision trees are easily parallelizable.

[10]This can become problematic when presenting the results to a client whose understanding of these matters is slight.

[11]Classical performance evaluation metrics can easily be fooled; if out of two classes one of the instances is only represented in 0.01% of the instances, predicting the non-rare class will yield correct predictions roughly 99.99% of the time, missing the point of the exercise altogether.

4.3 Performance Evaluation

As a consequence of the (so-called) **No-Free-Lunch Theorem**, no single classifier can be the best performer for every problem. Model selection must take into account:

- the **nature** of the available data;
- the **relative frequencies of the classification subgroups**;
- the **stated classification goals**;
- how easily the model lends itself to **interpretation** and **statistical analysis**;
- how much **data preparation** is required;
- whether it can accommodate various data types and missing observations;
- whether it performs well with large datasets, and
- whether it is **robust** against small data departures from theoretical assumptions.

Past success is not a guarantee of future success – it is the analyst's responsibility to try a variety of models. But how can the "best" model be selected?

When a classifier attempts to determine what kind of music a new customer would prefer, there is next to no cost in making a mistake; if, on the other hand, the classifier attempts to determine the presence or absence of cancerous cells in lung tissue, mistakes are more consequential.

Several metrics can be used to assess a classifier's performance, depending on the context.

Binary classifiers (such as the abstract example shown in Table 3) are simpler and have been studied far longer than multi-level classifiers; consequently, a larger body of evaluation metrics is available for these classifiers.

In the medical literature, TP, TN, FP and FN stand for **True Positives**, **True Negatives**, **False Positives**, and **False Negatives**, respectively. A perfect classifier would be one for which both $\text{FP}, \text{FN} = 0$. In practice, that rarely ever happens (if at all).

Traditional **performance metrics** include:

- sensitivity: TP/AP
- specificity: TN/AN
- precision: TP/PP
- negative predictive value: TN/PN
- false positive rate: FP/AN
- false discovery rate: $1-\text{TP}/\text{PP}$
- false negative rate: FN/AP
- accuracy: $(\text{TP}+\text{TN})/\text{T}$
- F_1–score: $2\text{TP}/(2\text{TP}+\text{FP}+\text{FN})$
- MCC:
$$\frac{\text{TP}\cdot\text{TN}-\text{FP}\cdot\text{FN}}{\sqrt{\text{AP}\cdot\text{AN}\cdot\text{PP}\cdot\text{PN}}}$$
- informedness/ROC: $\text{TP}/\text{AP}+\text{TN}/\text{AN}-1$
- markedness: $\text{TP}/\text{PP}+\text{TN}/\text{PN}-1$

The **confusion matrices** of two artificial binary classifers for a testing set are shown in Table 2. Both classifiers have an accuracy of 80%, but while the second classifier sometimes makes a wrong prediction for A, it never does so for B, whereas the first classifier makes erroneous predictions for both A and B.

On the other hand, the second classifier mistakenly predicts occurrence A as B 16 times while the first one only does so 6 times.

So which one is best? The performance metrics alone do not suffice to answer the question: the cost associated with making a mistake must also be factored in. Furthermore, it could be preferable to select performance evaluation metrics that generalize more readily to **multi-level classifiers** (see Table 4 for examples of associated confusion matrices).

Accuracy is the proportion of correct predictions amid all the observations; its value ranges from 0% to 100%. The higher the accuracy, the better the match, and yet, a predictive model with high accuracy may nevertheless be useless thanks to the **Accuracy Paradox** (see footnote, p. 16).

The **Matthews Correlation Coefficient** (MCC), on the other hand, is a statistic which is of use even when the classes are of very different sizes. As a correlation coefficient between the actual and predicted classifications, its range varies from -1 to 1.

If $\text{MCC}=1$, the predicted and actual responses are identical, while if $\text{MCC}=0$, the classifier performs no better than a random prediction ("flip of a coin").

It is also possible to introduce two non-traditional performance metrics (which are nevertheless well-known statistical quantities) to describe how accurately a classifier preserves the classification distribution (rather than how it behaves on an observation-by-observation basis):

- Pearson's χ^2: $\frac{1}{\text{T}}\left((\text{PP}-\text{AP})^2/\text{PP}+(\text{PN}-\text{AN})^2/\text{PN}\right)$
- Hist: $\frac{1}{\text{T}}\left(|\text{PP}-\text{AP}|+|\text{PN}-\text{AN}|\right)$

Note, however, that these are non-standard performance metrics. For a given number of levels, the smaller these quantities, the more similar the actual and predicted distributions.

For **numerical targets** y with predictions $\hat{y}$, the confusion matrix is not defined, but a number of classical performance evaluation metrics can be used on the testing set: the

- **mean squared** and **mean absolute errors**
$$\text{MSE}=\text{mean}\left\{(y_i-\hat{y}_i)^2\right\},\quad \text{MAE}=\text{mean}\{|y_i-\hat{y}_i|\};$$
- **normalized mean squared/mean absolute errors**
$$\text{NMSE}=\frac{\text{mean}\left\{(y_i-\hat{y}_i)^2\right\}}{\text{mean}\left\{(y_i-\overline{y})^2\right\}},$$
$$\text{NMAE}=\frac{\text{mean}\left\{|y_i-\hat{y}_i|\right\}}{\text{mean}\left\{|y_i-\overline{y}|\right\}};$$

		Predicted A	Predicted B	Total	
Actuals	A	54	10	64	79.0%
	B	6	11	17	21.0%
Total		60	21	81	
		74.1%	25.9%		

Classification Rates	
Sensitivity:	0.84
Specificity:	0.65
Precision:	0.90
Negative Predictive Value:	0.52
False Positive Rate:	0.35
False Discovery Rate:	0.10
False Negative Rate:	0.16

Performance Metrics	
Accuracy:	0.80
F1-Score:	0.87
Informedness (ROC):	0.49
Markedness:	0.42
M.C.C.:	0.46
Pearson's chi2:	0.01
Hist. Stat:	0.10

		Predicted A	Predicted B	Total	
Actuals	A	54	0	54	66.7%
	B	16	11	27	33.3%
Total		70	11	81	
		86.4%	13.6%		

Classification Rates	
Sensitivity:	1.00
Specificity:	0.41
Precision:	0.77
Negative Predictive Value:	1.00
False Positive Rate:	0.59
False Discovery Rate:	0.23
False Negative Rate:	0.00

Performance Metrics	
Accuracy:	0.80
F1-Score:	0.87
Informedness (ROC):	0.41
Markedness:	0.77
M.C.C.:	0.56
Pearson's chi2:	0.33
Hist. Stat:	0.40

Table 2. Performance metrics for two (artificial) binary classifiers.

		Predicted Category I	Predicted Category II	Total
Actuals	Category I	TP	FN	AP
	Category II	FP	TN	AN
Total		PP	PN	T

Table 3. A general binary classifier.

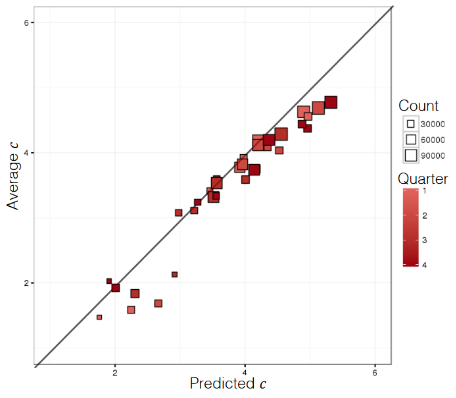


Figure 10. Predicted and actual responses (personal file).

- **mean average percentage error**

$$\text{MAPE} = \text{mean}\left\{\frac{|y_i - \hat{y}_i|}{y_i}\right\};$$

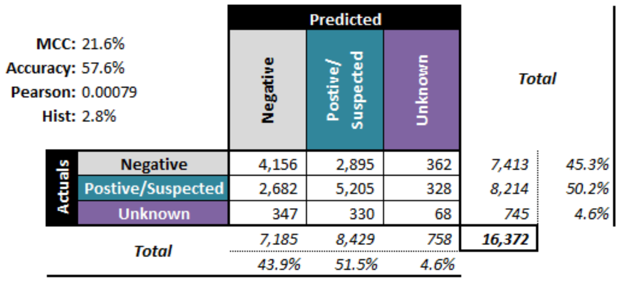
- **correlation** $\rho_{y,\hat{y}}$, which is based on the notion that for good models, the predicted values and the actual values should congregate around the lines $y = \hat{y}$ (see Figure 10 for an illustration).

As is the case for classification, an isolated value estimation performance metric does not provide enough of a rationale for model validation/selection. One possible exception: normalized evaluation metrics do provide some information about the relative quality of performance (see [21, 24]).

4.4 Toy Example: Kyphosis Dataset

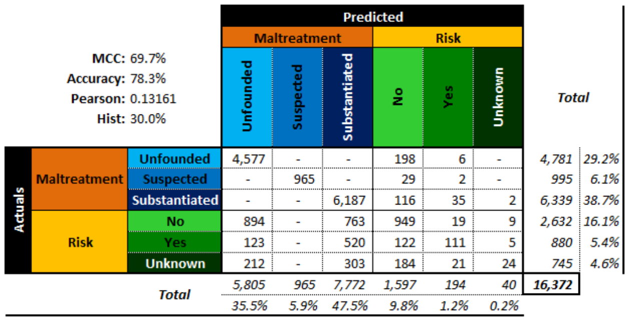
As a basic illustration of these concepts, consider the following example. **Kyphosis** is a medical condition related to an excessive convex curvature of the spine. Corrective spinal surgery is at times performed on children. A dataset of 81 observations and 4 attributes has been collected (we have no information on how the data was collected and how representative it is likely to be, but those details can be gathered from [7]). The attributes are:

- **kyphosis** (absent or present after presentation)
- **age** (at time of operation, in months)
- **number** (of vertebrae involved)
- **start** (topmost vertebra operated on)



MCC: 21.6%
Accuracy: 57.6%
Pearson: 0.00079
Hist: 2.8%

Actuals	Predicted: Negative	Postive/Suspected	Unknown	Total	
Negative	4,156	2,895	362	7,413	45.3%
Postive/Suspected	2,682	5,205	328	8,214	50.2%
Unknown	347	330	68	745	4.6%
Total	7,185	8,429	758	16,372	
	43.9%	51.5%	4.6%		



MCC: 69.7%
Accuracy: 78.3%
Pearson: 0.13161
Hist: 30.0%

Actuals		Predicted: Maltreatment: Unfounded	Suspected	Substantiated	Risk: No	Yes	Unknown	Total	
Maltreatment	Unfounded	4,577	-	-	198	6	-	4,781	29.2%
	Suspected	-	965	-	29	2	-	995	6.1%
	Substantiated	-	-	6,187	116	35	2	6,339	38.7%
Risk	No	894	-	763	949	19	9	2,632	16.1%
	Yes	123	-	520	122	111	5	880	5.4%
	Unknown	212	-	303	184	21	24	745	4.6%
Total		5,805	965	7,772	1,597	194	40	16,372	
		35.5%	5.9%	47.5%	9.8%	1.2%	0.2%		

Table 4. Performance metrics for (artificial) multi-level classifiers: ternary - left; senary - right (personal files).

The natural question of interest for this dataset is

> "how do the three explanatory attributes impact the operation's success?"

We use the `rpart` implementation of Classification and Regression Tree (CART) in R to generate a decision tree. Strictly speaking, this is not a predictive supervised task as we treat the entire dataset as a training set for the time being – there are no hold-out testing observations. The results are shown in Figure 11.

Interestingly, it would appear that the `number` variable does not play a role in determining the success of the operation (for the observations in the dataset).

Furthermore, the decision tree visualization certainly indicates that its leaves are not pure (see Figure 12).

Some additional (off-page) work suggests that the tree is somewhat overgrown and that it could benefit from being pruned after the first branching point (see Figure 12, again).

At any rate, it remains meangingless to discuss the performance of the tree for **predictive purposes** if we are not using a holdout testing sample (not to say anything about the hope of generalizing to a larger population).

To that end, we trained a model on 50 randomly selected observations and evaluated the performance on the remaining 31 observations (the structure of the tree is not really important at this stage). The results are shown in Table 5.

Is the model "good"? It is difficult to answer this question in the machine learning sense without being able to compare its performance metrics with those of other models (or families of models).[12]

In Section 6, we will briefly discuss how estimate a model's true predictive error rate through **cross-validation**. We will also discuss a number of other issues that can arise when ML/AI methods are not used correctly.

[12]The relative small size of the dataset should give consultants and data analysts pause for thought, at the very least.

4.5 Case Study: Minnesota Tax Audits

Large gaps between revenue owed (in theory) and revenue collected (in practice) are problematic for governments. Revenue agencies implement various fraud detection strategies (such as audit reviews) to bridge that gap.

Since business audits are rather costly, there is a definite need for algorithms that can predict whether an audit is likely to be successful or a waste of resources.

In *Data Mining Based Tax Audit Selection: A Case Study of a Pilot Project at the Minnesota Department of Revenue* [23], Hsu et al. study the Minnesota Department of Revenue's (DOR) tax audit selection process with the help of classification algorithms.

Objective The U.S. Internal Revenue Service (IRS) estimated that there were large gaps between **revenue owed** and **revenue collected** for 2001 and for 2006. Using DOR data, the authors sought to increase **efficiency** in the audit selection process and to **reduce the gap** between revenue owed and revenue collected.

Methodology The authors took the following steps:

1. **data selection and separation:** experts selected several hundred cases to audit and divided them into training, testing and validating sets;
2. **classification modeling** using MultiBoosting, Naïve Bayes, C4.5 decision trees, multilayer perceptrons, support vector machines, etc;
3. **evaluation of all models** was achieved by testing the model on the testing set – models originally performed poorly on the testing set until the size of the business being audited was recognized to have an effect, leading to two separate tasks (large and small businesses);
4. **model selection and validation** was done by comparing the estimated accuracy between different classification model predictions and the actual field audits. Ultimately, MultiBoosting with Naïve Bayes was selected as the final model; the combination also suggested some improvements to increase audit efficiency.

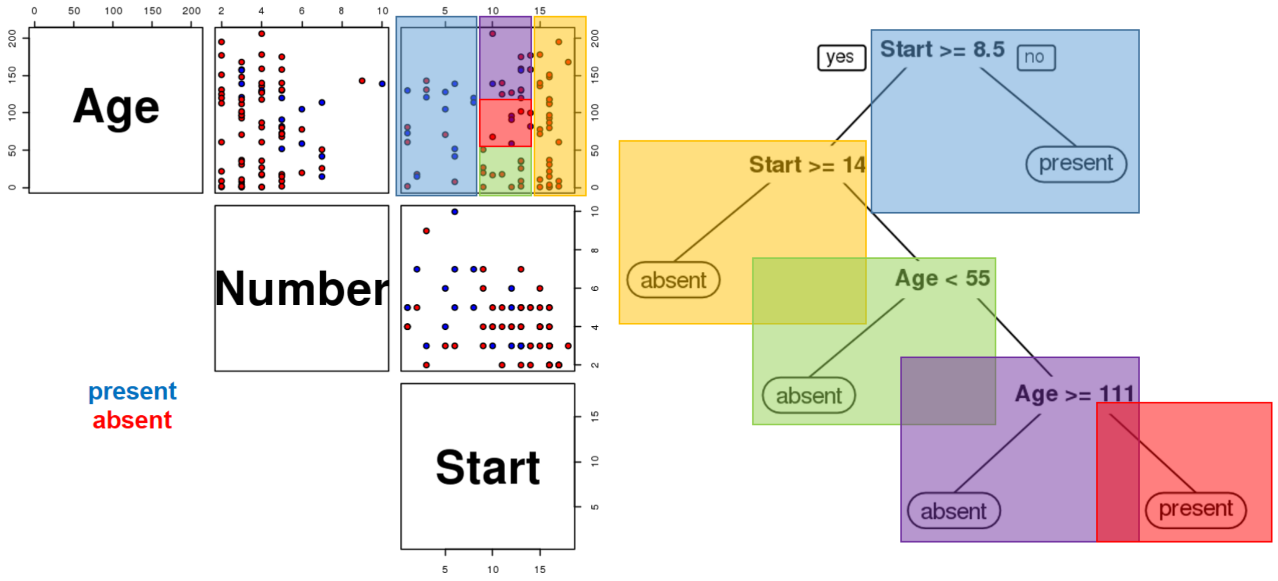


Figure 11. Kyphosis decision tree visualization. Only two features are used to construct the tree. We also note that the leaves are not pure – there are blue **and** red instances in 3 of the 5 classification regions.

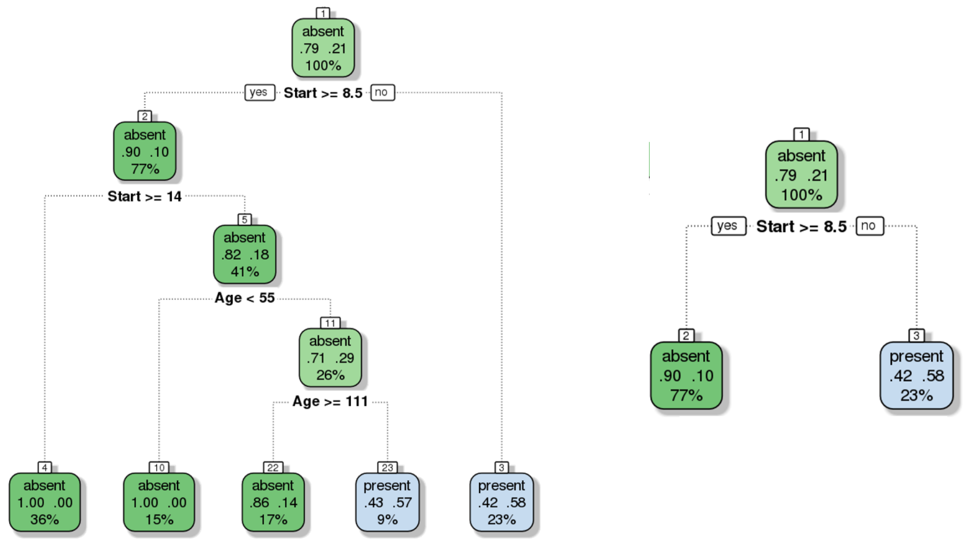
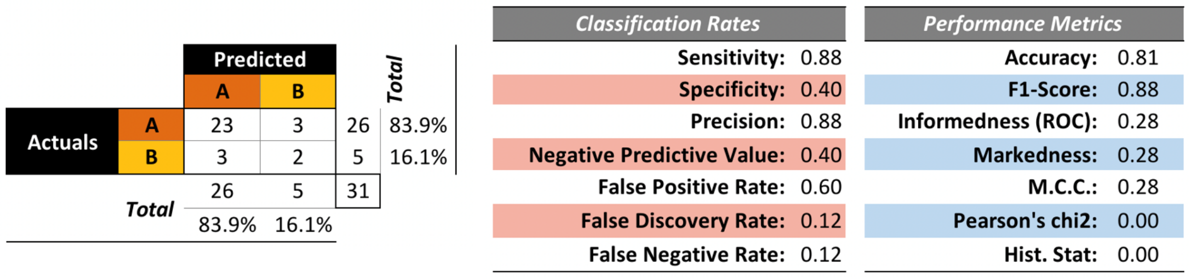


Figure 12. Pruning a decision tree – the original tree (left) is more accurate/more complex than the pruned tree (right).



		Predicted A	Predicted B	Total	
Actuals	A	23	3	26	83.9%
	B	3	2	5	16.1%
Total		26	5	31	
		83.9%	16.1%		

Classification Rates		Performance Metrics	
Sensitivity:	0.88	Accuracy:	0.81
Specificity:	0.40	F1-Score:	0.88
Precision:	0.88	Informedness (ROC):	0.28
Negative Predictive Value:	0.40	Markedness:	0.28
False Positive Rate:	0.60	M.C.C.:	0.28
False Discovery Rate:	0.12	Pearson's chi2:	0.00
False Negative Rate:	0.12	Hist. Stat:	0.00

Table 5. Kyphosis decision tree – performance evaluation. The accuracy and $F1$ score are good, but the false discovery rate and false negative rate are not so great. This tree is good at predicting successful surgeries, but not fantastic at predicting failed surgeries. Is it still useful?

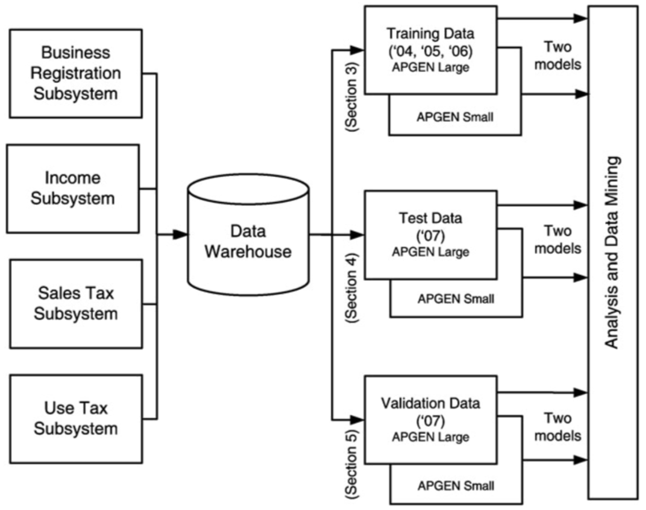


Figure 13. Data sources for APGEN mining [23]. Note the 6 final sets which feed the Data Analysis component.

Data The data consisted of selected tax audit cases from 2004 to 2007, collected by the audit experts, which were split into training, testing and validation sets:

- the **training data** set consisted of *Audit Plan General* (APGEN) *Use Tax* audits and their results for the years 2004-2006;
- the **testing data** consisted of APGEN Use Tax audits conducted in 2007 and was used to test or evaluate models (for Large and Smaller businesses) built on the training dataset,
- while **validation** was assessed by actually conducting field audits on predictions made by models built on 2007 Use Tax return data processed in 2008.

None of the sets had records in common (see Figure 13).

Strengths and Limitations of Algorithms

- The Naïve Bayes classification scheme assumes independence of the features, which rarely occurs in real-world situations. This approach is also known to potentially introduce bias to classification schemes. In spite of this, classification models built using Naïve Bayes have a successfully track record.
- MultiBoosting is an **ensemble technique** that uses committee (i.e. groups of classification models) and "group wisdom" to make predictions; unlike other ensemble techniques, it is different from other ensemble techniques in the sense that it forms a committee of sub-committees (i.e., a group of groups of classification models), which has a tendency to reduce both bias and variance of predictions (see [1,31] for more information on these topics).

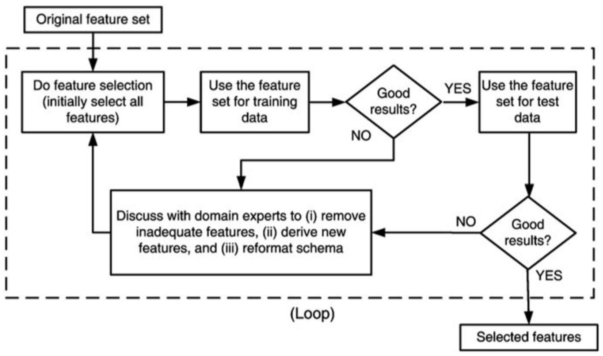

Original feature set
Do feature selection (initially select all features)
Use the feature set for training data
Good results?
YES
NO
Use the feature set for test data
Discuss with domain experts to (i) remove inadequate features, (ii) derive new features, and (iii) reformat schema
NO
Good results?
YES
(Loop)
Selected features

Figure 14. The feature selection process [23]. Note the involvement of domain experts.

Procedures Classification schemes need a response variable for prediction: audits which yielded more than $500 per year in revenues during the audit period were classified as *Good*; the others were *Bad*. The various models were tested and evaluated by comparing the performances of the manual audits (which yield the actual revenue) and the classification models (the predicted classification).

The procedure for manual audit selection in the early stages of the study required:

1. DOR experts selecting several thousand potential cases through a query;
2. DOR experts further selecting several hundreds of these cases to audit;
3. DOR auditors actually auditing the cases, and
4. calculating audit accuracy and return on investment (ROI) using the audits results.

Once the ROIs were available, data mining started in earnest. The steps involved were:

1. **Splitting the data** into training, testing, and validating sets.
2. **Cleaning the training data** by removing "bad" cases.
3. **Building** (and revising) **classification models** on the training dataset. The first iteration of this step introduced a separation of models for larger businesses and relatively smaller businesses according to their **average annual withholding amounts** (the threshold value that was used is not revealed in [23]).
4. **Selecting separate modeling features** for the APGEN Large and Small training sets. The feature selection process is shown in Figure 14.
5. **Building classification models** on the training dataset for the two separate class of business (using C4.5, Naïve Bayes, multilayer perceptron, support vector machines, etc.), and assessing the classifiers using **precision** and **recall** with improved estimated ROI:

$$\text{Efficiency} = \text{ROI} = \frac{\text{Total revenue generated}}{\text{Total collection cost}}$$

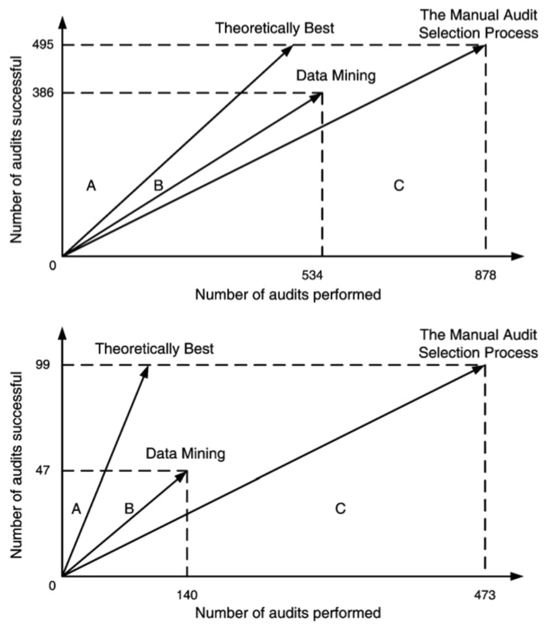


Figure 15. Audit resource deployment efficiency [23]. Top: APGEN Large (2007). Bottom: APGEN Small (2007). In both cases, the Data Mining approach was more efficient (the slope of the Data Mining vector is "closer" to the Theoretical Best vector than is the Manual Audit vector).

Results, Evaluation and Validation The models that were eventually selected were combinations of MultiBoosting and Naïve Bayes (C4.5 produced interpretable results, but its performance was shaky).

For APGEN Large (2007), experts had put forward 878 cases for audit (495 of which proved successful), while the classification model suggested 534 audits (386 of which proved successful). The theoretical best process would find 495 successful audits in 495 audits performed, while the manual audit selection process needed 878 audits in order to reach the same number of successful audits.

For APGEN Small (2007), 473 cases were recommended for audit by experts (only 99 of which proved successful); in contrast, 47 out of the 140 cases selected by the classification model were successful. The theoretical best process would find 99 successful audits in 99 audits performed, while the manual audit selection process needed 473 audits in order to reach the same number of successful audits.

In both cases, the classification model improves on the manual audit process: roughly 685 data mining audits to reach 495 successful audits of APGEN Large (2007), and 295 would be required to reach 99 successful audits for APGEN Small (2007), as can be seen in Figure 15.

Table 6 presents the confusion matrices for the classification model on both the APGEN Large and Small 2007 datasets.

The revenue R and collection cost C entries can be read as follows: the 47 successful audits which were correctly

	Predicted as good	Predicted as bad
Actually good	386 (Use tax collected)	109 (Use tax lost)
	R = $5,577,431 (83.6 %)	R = $925,293 (13.9 %)
	C = $177,560 (44 %)	C = $50,140 (12.4 %)
Actually bad	148 (costs wasted)	235 (costs saved)
	R = $72,744 (1.1 %)	R = $98,105 (1.4 %)
	C = $68,080 (16.9 %)	C = $108,100 (26.7 %)

	Predicted as good	Predicted as bad
Actually good	47 (Use tax collected)	52 (Use tax lost)
	R = $263,706 (42.5 %)	R = $264,101 (42.5 %)
	C = $21,620 (9.9 %)	C = $23,920 (11 %)
Actually bad	93 (costs wasted)	281 (costs saved)
	R = $24,441 (3.9 %)	R = $68,818 (11.1 %)
	C = $42,780 (19.7 %)	C = $129,260 (59.4 %)

Table 6. Confusion matrices for audit evaluation [23]. Top: APGEN Large (2007). Bottom: APGEN Small (2007). R stands for revenues, C for collection costs.

identified by the model for APGEN Small (2007) correspond to cases consuming 9.9% of collection costs but generating 42.5% of the revenues. Similarly, the 281 bad audits correctly predicted by the model represent notable collection cost savings. These are associated with 59.4% of collection costs but they generate only 11.1% of the revenues.

Once the testing phase of the study was conpleted, the DOR validated the data mining-based approach by using the models to select cases for actual field audits in a real audit project. The prior success rate of audits for APGEN Use tax data was 39% while the model was predicting a success rate of 56%; the actual field success rate was 51%.

Take-Aways A substantial number of models were churned out before the team made a final selection. Past performance of a model family in a previous project can be used as a guide, but it provides no guarantee regarding its performance on the current data – remember the *No Free Lunch (NFL) Theorem* [54]: nothing works best all the time!

There is a definite iterative feel to this project: the feature selection process could very well require a number of visits to domain experts before the feature set yields promising results. This is a valuable reminder that the data analysis team should seek out individuals with a good understand of both data and context. Another consequence of the NFL is that domain-specific knowledge has to be integrated in the model in order to beat random classifiers, on average [53].

Finally, this project provides an excellent illustration that even slight improvements over the current approach can find a useful place in an organization – data science is not solely about Big Data and disruption!

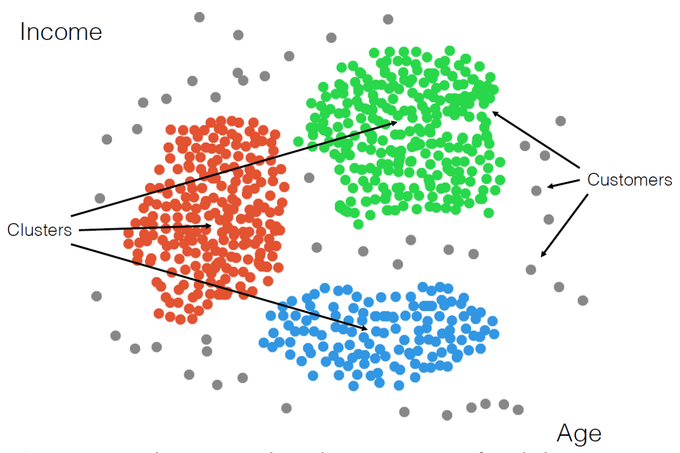


Figure 16. Clusters and outliers in an artificial dataset.

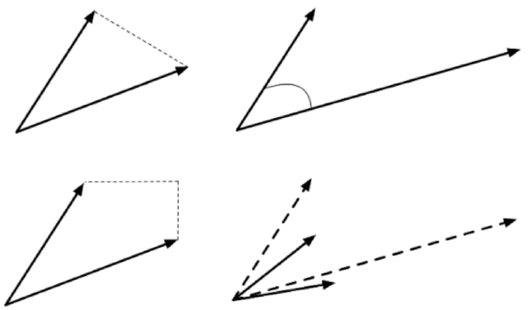


Figure 17. Distance metrics between observations: Euclidean (as the crow flies, top left); cosine (direction from a vantage point, top right); Manhattan (taxi-cab, bottom left). Observations should be transformed (scaled, translated) before distance computations (bottom right).

5. Unsupervised Learning and Clustering

A Never-Ending Story

Clustering is in the eye of the beholder, and as such, researchers have proposed many induction principles and models whose corresponding optimisation problem can only be approximately solved by an even larger number of algorithms.

– V. Estivill-Castro
Why So Many Clustering Algorithms?

We can make a variety of **quantitative statements** about a dataset, at the **univariate** level. For instance, we can

- compute **frequency counts** for the variables, and
- identify **measures of centrality** (mean, mode, median), and
- **dispersal** (range, standard deviation), among others.

At the **multivariate** level, the various options include *n*—**way tabulations**, **correlation analysis**, and **data visualization**, among others.

While these might provide insights in simple situations, datasets with a **large number of variables** or with **mixed types** (categorical and numerical) might not yield to such an approach. Instead, insights might come in the form of **aggregation** or **clustering** of similar observations (see Figure 16).

A successful **clustering scheme** is one that tightly joins together any number of similarity profiles ("tight" in this context refers to small variability within the cluster).

A typical application is one found in **search engines**, where the listed search results are the nearest similar objects (**relevant webpages**) clustered around the search item.

Dissimilar objects (**irrelevant webpages**) should not appear in the list, being "far" from the search item.

Left undefined in this example is the crucial notion of **closeness**: what does it mean for one observation to be near another one? Various **metrics** can be used (see Figure 17), and not all of them lead to the same results.

Clustering is a form of **unsupervised learning** since the cluster labels (and possibly their number) are not determined ahead of the analysis.

The algorithms can be **complex** and **non-intuitive**, based on varying notions of similarities between observations, and yet, the temptation to provide a simple *a posteriori* explanation for the groupings remains **strong** – we really, really want to reason with the data.[13]

The algorithms are also (typically) **non-deterministic** – the same routine, applied twice (or more) to the same dataset, can discover completely different clusters.[14]

This (potential) non-replicability is not just problematic for validation – it can also leads to client dissatisfaction. If the consultant is tasked with finding customer clusters for marketing purposes and the clusters change every time the client asks for a report, they will be very confused (and doubtful) unless the **stochastic nature** of the process has already been explained.

Another interesting aspect of clustering algorithms is that they often find clusters when there are no natural ways to break down a dataset into constituent parts.

Indeed, on the one hand, if there is no natural way to break up the data into clusters, the results may be **arbitrary** and fail to represent any underlying reality of the dataset.

On the other, it could be that while there was no **recognized** way of naturally breaking up the data into clusters, the algorithm **discovered** such a grouping – clustering is sometimes called **automated classification** as a result.

[13]Were you able to look at Figure 16 without assigning labels or trying to understand what type of customers were likely to be young and have medium income? Older and wealthier?

[14]The order in which the data is presented can play a role, as can starting configurations.

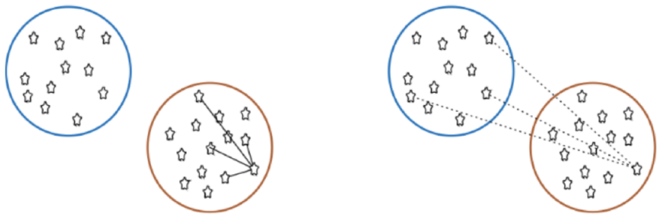


Figure 18. Distance to points in own clusters (left, smaller is better) and to points in other clusters (right, larger is better).

The aim of clustering, then, is to divide into **naturally occurring groups**. Within each group, observations are similar; between groups, they are dissimilar (see Figure 18).

As a learning process, clustering is fairly **intuitive** for human beings – our brains unconsciously search for patterns and they can generally handle **messy** data with the same relative ease as clean data.

Computers have a harder time of it, however; part of the challenge is that there is no **agreed-upon definition of what constitutes a cluster**, and so we cannot easily code their recognition into algorithms – to paraphrase Justice Potter Stewart,

"I may not be able to define what a cluster is, but I know one when I see one."

All clustering algorithms rely on the notion of **similarity** w between observations; in many instances, similarity is obtained *via* a **distance** (or metric) d, with $w \to 1$ as $d \to 0$, and $w \to 0$ as $d \to \infty$.

However, there exist similarity measures which are not derived from a distance metric.

One additional clustering challenge is that there is no such thing as **the** distance or **the** similarity measure between observations – observations which are similar using a specific measure **may not be similar at all using another**.

Commonly-used metrics include:

- euclidean,
- Manhattan,
- cosine,
- Canberra,
- Haming,
- Jaccard,
- Pearson,
- and so forth.

Note, however, that no matter which similarity measure is selected, the data must first be **transformed**: scaled, centered, etc. (see Figure 17). This introduces another layer of arbitrary choices, as there are multiple available options and no **canonical** way to perform this.

Applications Frequently, we use clustering and other unsupervised learning tasks as **preliminary steps** in supervised learning problems, but there exist stand-alone applications as well:

- **text analysis** – grouping similar documents according to their topics, based on the patterns of common and unusual terms;
- **product recommendations** – grouping online purchasers based on the products they have viewed, purchased, liked, or disliked, or grouping products based on customer reviews;
- **marketing** – grouping client profiles based on their demographics and preferences;
- **social network analysis** – recognising communities within large groups of people;
- **medical imaging** – differentiating between different tissue types in a 3D voxel;
- **genetics** – inferring structures in populations;
- dividing a larger group (or area, or category) into smaller groups, with members of the smaller groups having similarities of some kind, as analytical tasks may then be solved separately for each of the smaller groups, which may lead to increased accuracy once the separate results are aggregated, or
- creating (new) taxonomies on the fly, as new items are added to a group of items, which could allow for easier product navigation on a website like Netflix, for instance.

Other examples may be found in [2,4,10,13,18,25,30,37–39,41,46,47].

When all is said and done, the clustering process is quite standard, notwithstanding the choice of scaling strategy, similarity measure, and algorithm and parameters (see the pipeline shown in Figure 19).

5.1 Clustering Algorithms

As is the case with classification, the number of clustering algorithms is quite high; the Wikipedia page lists 40+ such algorithms as of August 2018 [51]. The choice of algorithms (and associated parameters) is as much an art as it is a science, although domain expertise can come in handy [2].

There is a smaller list of common algorithms that data scientists and consultants should have in their toolbox (full descriptions available in [2,42,50]):

- $k-$**means**, close on the heels of decision trees for the title of "most-used data science algorithm", is a **partition clustering method** which tends to produce equal-sized clusters; when clients ask for their data to be clustered, they are typically envisioning $k-$means with the Euclidean metric; variants include $k-$mode (for categorical data), $k-$medians (for data

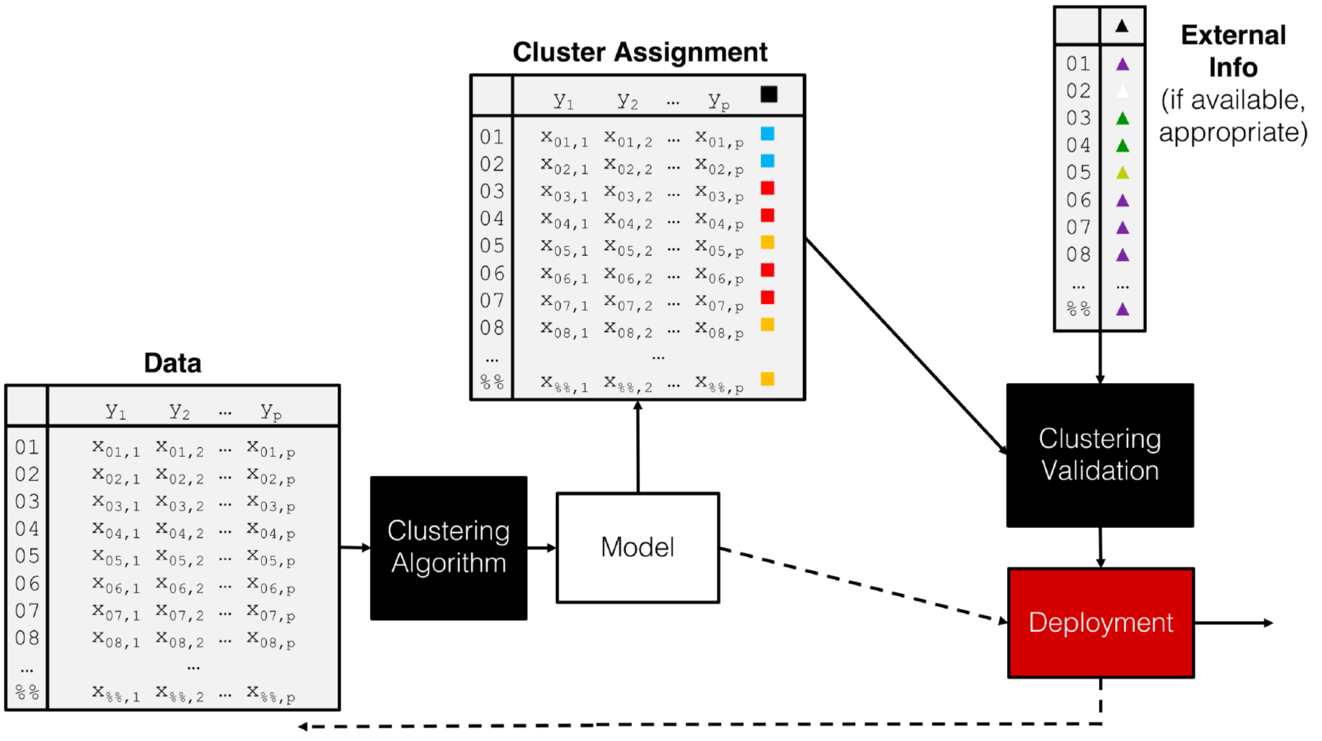


Figure 19. A clustering pipeline, including validation and (eventual) deployment.

with outliers), and k—means|| and k—means++ for large data sets; the number of clusters k (and the similarity measure/distance metric) must be provided by the user; works fairly well for "blob"-like data;

- **hierarchical clustering** is one of the few deterministic algorithms on the market, with **divisive** (DIANA) and **agglomerative** (AGNES) versions; no parameters need to be inputted, but the users must select a **linkage** strategy (roughly speaking, a metric that computes the distance between clusters) and a level at which to read off the clusters (see Figure 20);
- **density-based spatial clustering** (DBSCAN) is a graph-based approach which attempts to identify densely-packed regions in the dataset; its most obvious advantages (and of its variants OPTICS and DENCLUE) are **robustness to outliers** and not needing to input a **number of clusters** to search for in the data; the main disadvantage is that the optimal input parameters (**neighbourhood radius** and **minimum number of points** to be considered dense) are not easy to derive;
- **affinity propagation** is another algorithm which selects the optimal number of clusters directly from the data, but it does so by trying and evaluating various scenarios, which may end up being **time-consuming**;
- **spectral clustering** can be used to recognise **non-globular** clusters; these are found by computing eigenvalues of an associated Laplacian matrix – consequently, spectral clustering is fast.

Other methods include **latent Dirichlet allocation** (used in topics modeling), **expectation-maximisation** (particularly useful to find gaussian clusters), **BIRCH** (a local method which does not require the entire dataset to be scanned) and **fuzzy clustering** (a soft clustering scheme in which the observations have a degree of belonging to each cluster).

5.2 k—Means

As mentioned previously, k—means is a very natural way to group observations together (formally, k—means is linked to **Voronoi tilings**).

k—means clustering is achieved by:

1. selecting a **distance metric** d (based on the data type and domain expertise);
2. selecting a **number of clusters** k;
3. randomly choosing k data instances as initial **cluster centres**;
4. calculating the **distance** from each observation to each centre;
5. placing each instance in the cluster whose centre it is **nearest** to;
6. computing/updating the **centroid** for each cluster (see Figure 21);
7. repeating steps 4-6 until the clusters are **stable**.

For k—means, cluster centroids are obtained by averaging all points in the cluster. For k—medians and k—mode, the centrality measure is replaced by the obvious candidate.

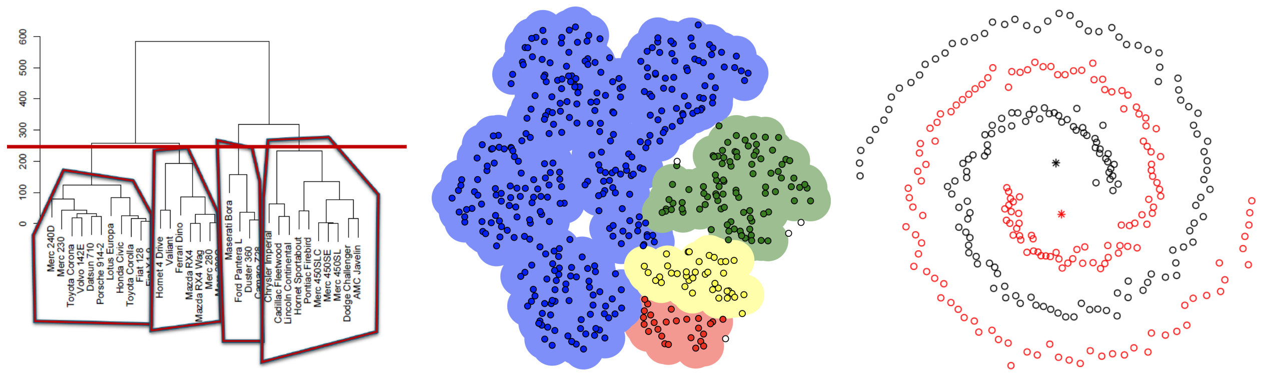


Figure 20. Illustration of hierarchical clustering (left), DBSCAN (middle, based on [20]), and spectral clustering (right).

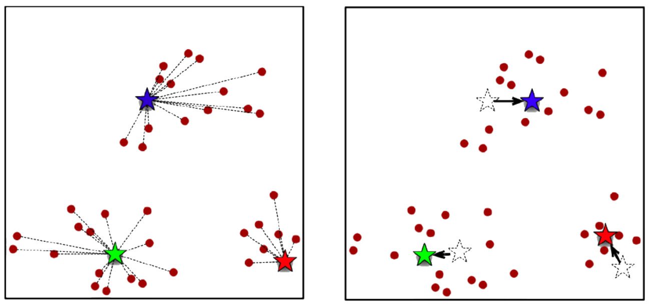


Figure 21. k—means cluster allocation (left) and updated centres (right).

This simple algorithm has numerous strengths:

- it is elegant and **easy to implement** (without actually having to compute pairwise distances), and so is extremely common as a result;
- in many contexts, it is a **natural way** to look at grouping observations, and
- it helps provide a first-pass **basic understanding of the data structure**.

On the other hand,

- it can only assign an instance to **one** cluster, which can lead to overfitting – a more robust solution would be to compute the probability of belonging to each cluster, perhaps based on the distance to the centroid;
- it requires the "true" underlying clusters to be **gaussian-** or blob-shaped, and it will fail to to produce useful clusters if that assumption is not met in practice,
- it does not allow for **overlapping** or **hierarchical** groupings.

Notes Let us now return to some issues relating to clustering **in general** (and not just to k—means):

No matter the choice of algorithm, clustering rests on the assumption that **nearness of observations** (in whatever metric) is linked with **object similarity** similarity, and that **large distances** are linked with **dissimilarity**.

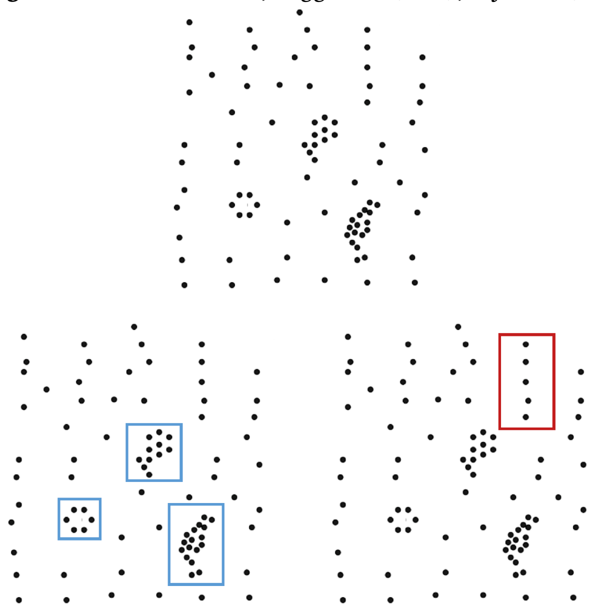
While there are plenty of situations where this is an appropriate assumption to make (temperature readings on a map, for instance), there are others where it is unlikely to be the case (chocolate bars and sensationalist tabloids at a grocery's checkout, say).

The **lack of a clear-cut definition** of what a cluster actually is (see Figure 22) makes it difficult to validate clustering results. Much more can be said on the topic [2].

The fact that various algorithms are **non-deterministic** is also problematic – clustering schemes should never be obtained using **only one** algorithmic pass, as the outcome could be different depending on the **location** of random starting positions and the distance/similarity metric in use.

But this apparent fickleness is not necessarily a problem: **essential patterns** may emerge if the algorithms are implemented multiple times, with different starting positions and re-ordered data (see **cluster ensembles** [2]).

Figure 22. Artificial data; suggested (blue), rejected (red).



For those algorithms that require the **number of clusters** as an input, it is difficult to determine what the optimal number should be (see Figure 23).

This number obviously depends on the choice of algorithm/metric, the underlying data, and the use that will be made of the resulting clusters – a dataset could have 3 natural groups when seen through the lens of k—means, but only 2 clusters for a specific choice of parameter values in DBSCAN, and so on.

This problem could be overcome by producing clustering schemes (from the same family of algorithms) with an increasing number of clusters and to plot the average distance of a cluster member to its cluster representative (centroid) against the number of clusters. Any kink in the plot represents a number of clusters at which an increase does not provide an in-step increase in clustering "resolution", so to speak (see Figure 26 for an example).

Even when a cluster scheme has been accepted as valid, a **cluster description** might be difficult to come by – should clusters be described using representative instances or average values or some combination of its' members most salient features?

Although there are exceptions, the ease with which clusters can be described often provides an indication about how **natural** the groups really are.

One of the most frustrating aspects of the process is that most methods will find clusters in the data even if there are none – although DBSCAN is exempt from this **ghost clustering** (see Figure 24).

Finally, consultants and analysts should beware the temptation of ***a posteriori* rationalisation** – once clusters have been found, it is tempting to try to "explain" them. Why are the groups as they have been found?

But that is a job for domain experts, at best, and a waste of time and resources, at worst. Thread carefully.

5.3 Clustering Validation

What does it mean for a clustering scheme to be **better** than another? What does it mean for a clustering scheme to be **valid**? What does it mean for a single cluster to be **good**? **How many** clusters are there in the data, really?

These are not easy questions to answer. In general, asking if a clustering scheme is the right one or a good one is meaningless – much better to ask if it is **optimal** or **sub-optimal**, potentially in comparison to other schemes.

An **optimal** clustering scheme is one which

- **maximizes separation** between clusters;
- **maximizes similarity** within groups;
- agrees with the **human eye test**, and
- is **useful** at achieving its goals.

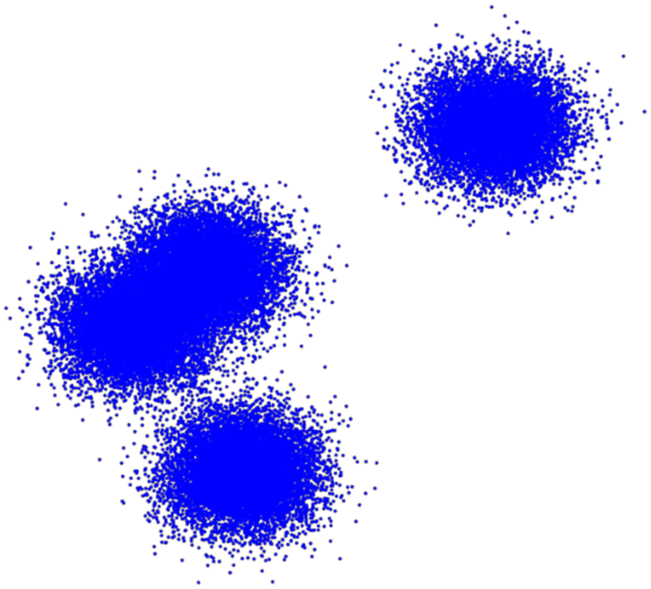


Figure 23. The number of clusters in a dataset is ambiguous: are there 2, 3, 4+ clusters?

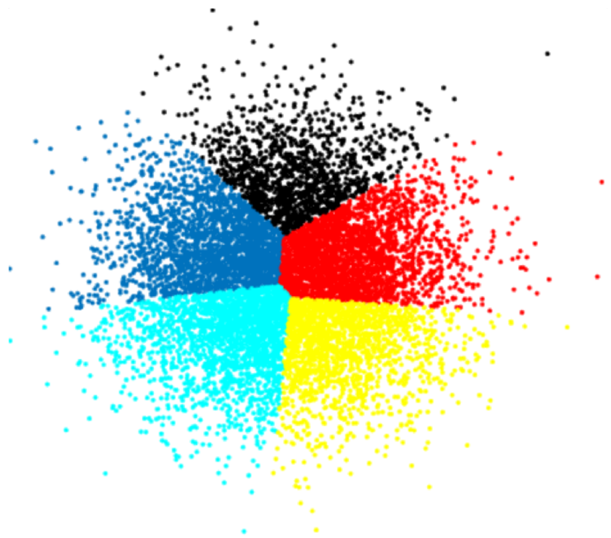


Figure 24. An illustration of ghost clustering with k—means, for $k = 5$.

There are 3 families of **clustering validation** approaches:

- **external**, which use additional information (but the labels in question might have very little to do with the similarity of the observations);
- **internal**, which use only the clustering results (shape and size of clusters, etc), and
- **relative**, which compare across a number of clustering attempts.

In order to illustrate some of the possibilities, consider a dataset with **clustering scheme** $\mathscr{C} = \{\mathscr{C}_1, \ldots, \mathscr{C}_N\}$, where $\mathscr{C}_m$'s centroid is denoted by c_m, and the average distance of $\mathscr{C}_m$'s members to c_m is denoted by s_m.

The **Davies-Bouldin Index** is defined as

$$\text{DB}_{\mathscr{C}} = \frac{1}{N} = \sum_{i=1}^{N} \max_{j\neq i} \left\{ \frac{s_i + s_j}{d(c_i, c_j)} \right\},$$

where d is the selected distance metric. Since $\text{DB}_{\mathscr{C}}$ is only defined using the clustering results, it is an **internal validation method**.

Heuristically, if the **cluster separation is small**, we might expect $d(c_i, c_j)$ to be (relatively) small, and so $\text{DB}_{\mathscr{C}}$ should be (relatively) large.

In the same vein, if the clusters are **heterogeneous**, we might expect $s_i + s_j$ to be (relatively) large, and so $\text{DB}_{\mathscr{C}}$ should be (relatively) large.

In short, when the clustering scheme is **sub-optimal**, $\text{DB}_{\mathscr{C}}$ is "large". This suggests another way to determine the optimal number of clusters – pick the scheme with minimal $\text{DB}_{\mathscr{C}}$ (see Figure 26, using a modified version of the index).

Other **cluster quality metrics** exist, including **SSE**, **Dunn's Index**, **Silhouette Metric**, etc. [2, 12].

5.4 Toy Example: Iris Dataset

Iris is a genus of plants with showy flowers. The iris dataset contains 150 observations of 5 attributes for specimens collected by Anderson, mostly from a Gaspé peninsula's pasture in the 1930s [16]. The attributes are

- **petal width**
- **petal length**
- **sepal width**
- **sepal length**
- **species** (virginica, versicolor, setosa)

A "description" of these features is provided by the picture in Figure 25 (left).

This dataset has become synonymous with data analysis,[15] being used to showcase just about every algorithm under the sun. This is, sadly, also what we are going to do.[16]

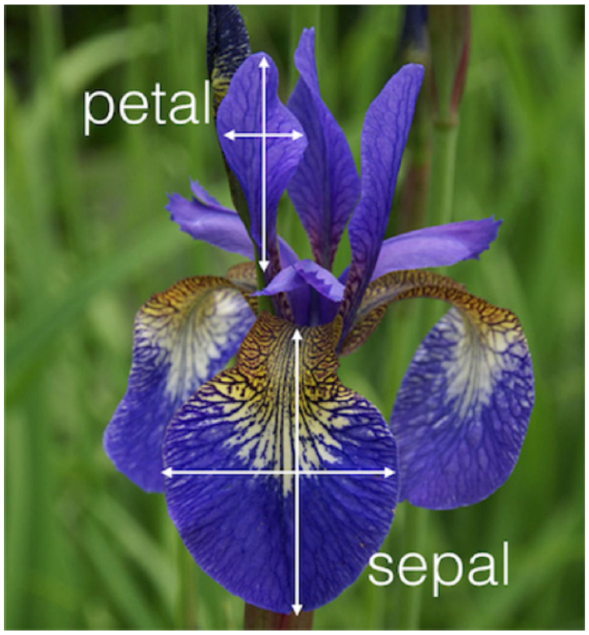
A **principal component projection** of the dataset, with species indicated by colours, is shown in Figure 25.

From an **unsupervised learning** point of view, one question of interest is whether the observations form natural groupings, and, if so, whether these groupings correspond to the (known) species.

We use the k—means algorithm with Euclidean distance to resolve the problem. Since we do not know how many clusters there should be in the data (the fact that there are 3 species does not mean that there should be 3 clusters), we run 40 replicates for $k = 2, \ldots, 15$.

[15]To the point that the standard joke is that "it's not necessary to be a gardener to become a data analyst, but it helps".

[16]Note that the iris dataset has started being phased out in favour of the penguin dataset [43].



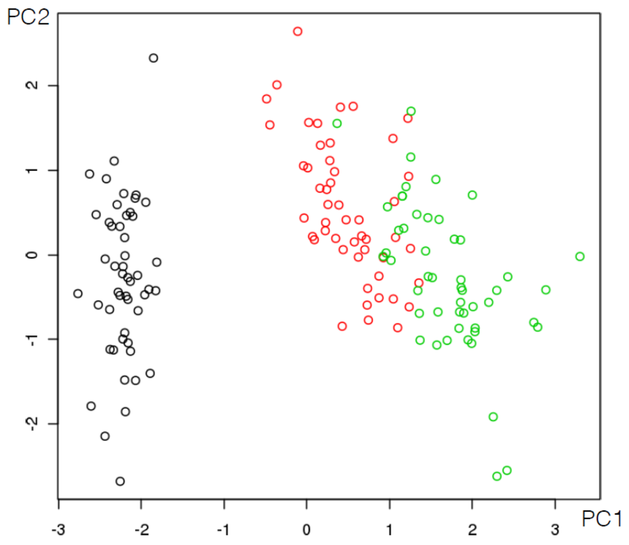


Figure 25. Illustration of iris measurements (top) and classification with 3 species, projected on the first 2 principal components (bottom).

The resulting clustering scheme for $k = 2, 3, 4, 15$ (for one of the replicates in each case) are shown in Figure 27.

For each replicate, we compute a (modified) **Davies-Bouldin Index** and the **Sum of Squared Errors** of the associated clustering schemes (see Figure 26) – the validation curves seem to indicate that there could be either 3 of 5 natural k—means clusters in the data. Is this a surprising outcome?

A single replicate with $k = 5$ is shown in Figure 28. Would you consider this representative final clustering scheme to be meaningful?

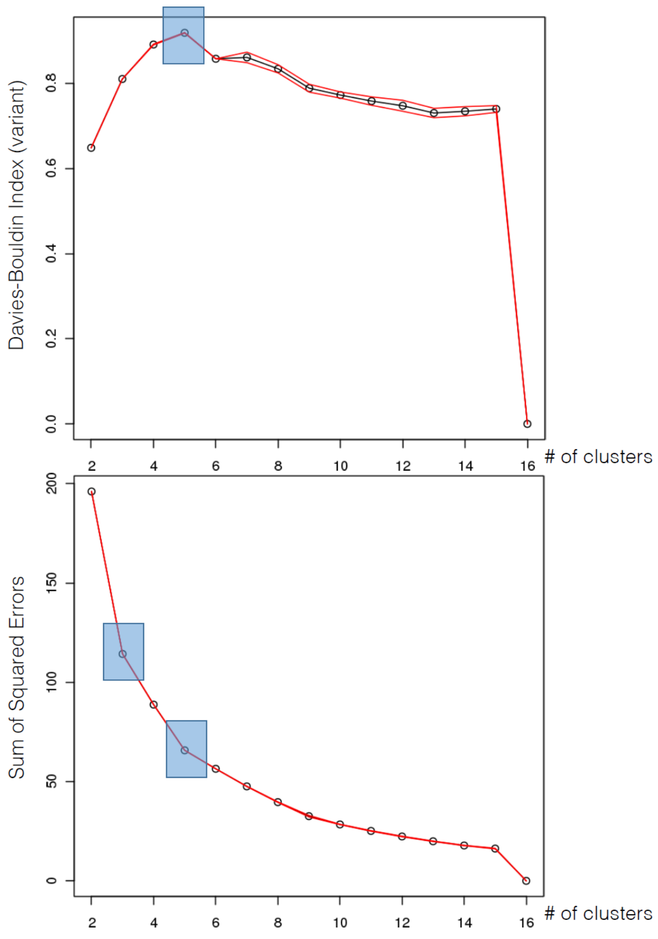


Figure 26. Optimal clustering results for the iris dataset: 5 clusters using (modified) Davies-Bouldin index and Sum of Squared Errors.

5.5 Case Study: The Livehoods Project

When we think of similarity at the urban level, we typically think in terms of neighbourhoods. Is there some other way to identify similar parts of a city?

In *The Livehoods Project: Utilizing Social Media to Understand the Dynamics of a City* [10], Cranshaw et al. study the social dynamics of urban living spaces with the help of clustering algorithms.

Objective The researchers aims to draw the boundaries of **livehoods**, areas of similar character within a city, by using clustering models. Unlike **static** administrative neighborhoods, the livehoods are defined based on the habits of people who live there.

Methodology The case study introduces **spectral clustering** to discover the **distinct geographic areas** of the city based on its inhabitants' collective **movement patterns**. Semi-structured interviews are also used to **explore**, **label**, and **validate** the resulting clusters, as well as the urban dynamics that shape them.

Livehood clusters are built and defined using the following methodology:

1. a **geographic distance** is computed based on pairs of check-in venues' coordinates;
2. **social similarity** between each pair of **venues** is computed using cosine measurements;
3. spectral clustering produces **candidate livehoods** clusters;
4. interviews are conducted with residents in order to **validate** the clusters discovered by the algorithm.

Data The data comes from two sources, combining approximately 11 million Foursquare (a recommendation site for venues based on users' experiences) check-ins from the dataset of Chen et al. [8] and a new dataset of 7 million Twitter check-ins downloaded between June and December of 2011.

For each check-in, the data consists of the **user ID**, the **time**, the **latitude and longitude**, the **name of the venue**, and its **category**.

In this case study, it is livehood clusters from the city of Pittsburgh, Pennsylvania, that are examined *via* 42,787 check-ins of 3840 users at 5349 venues.

Strengths and Limitations of the Approach

- The technique used in this study is **agnostic** towards the particular source of the data: it is not dependent on meta-knowledge about the data.
- The algorithm may be prone to "majority" bias, consequently misrepresenting/hiding minority behaviours.
- The dataset is built from a **limited** sample of check-ins shared on Twitter and are therefore biased towards the types of visits/locations that people typically want to share **publicly**.
- Tuning the clusters is non-trivial: experimenter bias may combine with "confirmation bias" of the interviewees in the validation stage – if the researchers are themselves residents of Pittsburgh, will they see clusters when there are none?

Procedures The Livehoods project uses a **spectral clustering model** to provide structure for local **urban areas** (UAs), grouping close Foursquare venues into clusters based on both the **spatial proximity** between venues and the **social proximity** which is derived from the distribution of people that check-in to them.

The guiding principle of the model is that the "character" of an UA is defined both by the types of venues it contains and by the people frequent them as part of their daily activities. These clusters are referred to as **Livehoods**, by analogy with more traditional neighbourhoods.

Let V be a list of Foursquare venues, A the associated **affinity matrix** representing a measure of similarity between each venue, and $G_m(A)$ be the graph obtained from the A by linking each venue to its nearest m neighbours.

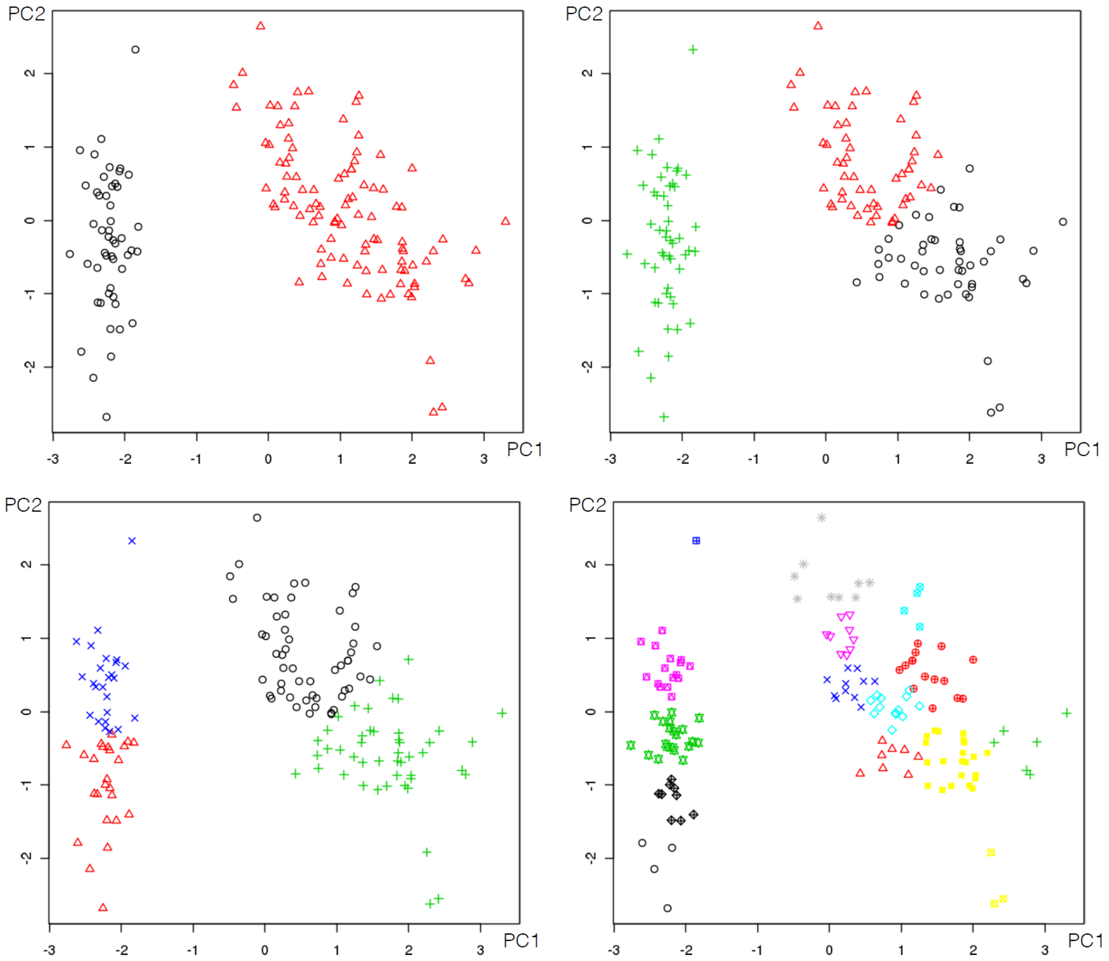


Figure 27. Clustering results on the iris dataset with k—means, for $k = 2, 3, 4, 15$ (from top left to bottom right, by row).

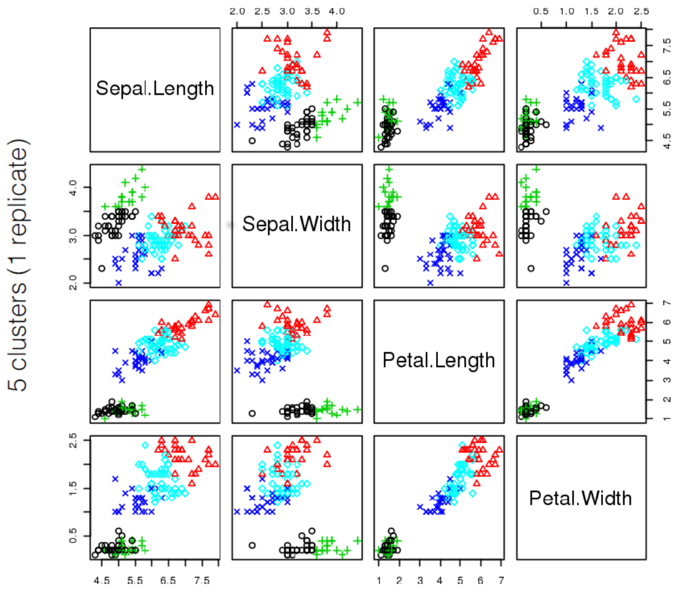


Figure 28. Optimal clustering results for the iris dataset (one replicate, $k = 5$).

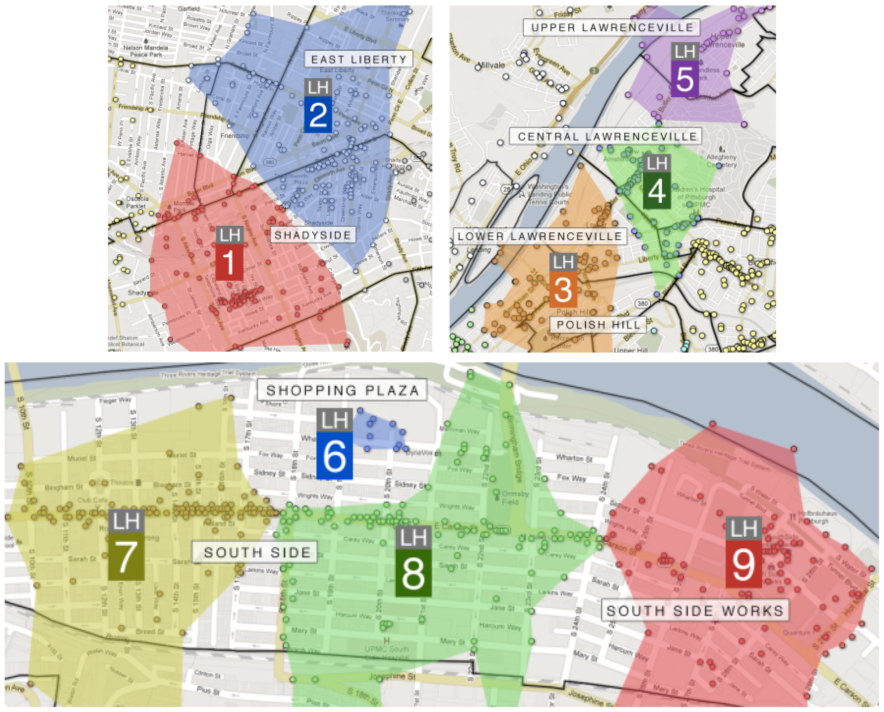


Figure 29. Some livehoods in metropolitan Pittsburgh, PA: Shadyside/East Liberty, Lawrenceville/Polish Hill, and South Side. Municipal borders are shown in black.

Spectral clustering is implemented as follows:

1. Compute the diagonal degree matrix $D_{ii} = \sum_j A_{ij}$;
2. Set the Laplacian matrix $L = D - A$ and
$$L_{\text{norm}} = D^{-1/2} L D^{-1/2};$$
3. Find the k smallest eigenvalues of L_{norm}, where k is the index which provides the biggest jump in successive eigenvalues of eigenvalues of L_{norm}, in increasing order;
4. Find the eigenvectors $e_1, \ldots e_k$ of L corresponding to the k smallest eigenvalues;
5. Construct the matrix E with the eigenvectors $e_1, \ldots e_k$ as columns;
6. Denote the rows of E by $y_1, \ldots, y_n$, and cluster them into k clusters $C_1, \ldots, C_k$ using k-means. This induces a clustering $\{A_1, \ldots, A_k\}$ defined by
$$A_i = \{j \mid y_j \in C_i\}.$$
7. For each A_i, let $G(A_i)$ be the subgraph of $G_m(A)$ induced by vertex A_i. Split $G(A_i)$ into connected components. Add each component as a new cluster to the list of clusters, and remove the subgraph $G(A_i)$ from the list.
8. Let b be the area of bounding box containing coordinates in the set of venues V, and b_i be the area of the box containing A_i. If $\frac{b_i}{b} > \tau$, delete cluster A_i, and redistribute each of its venues $v \in A_i$ to the closest A_j under the distance measurement.

Results, Evaluation and Validation The parameters used for the clustering were $m = 10$, $k_{\text{min}} = 30$, $k_{\text{max}} = 45$, and $\tau = 0.4$. The results for three areas of the city are shown in Figure 29. In total, 9 livehoods have been identified and validated by 27 Pittsburgh residents (see Figure 29; the original report has more information on this process).

- **Municipal Neighborhoods Borders:** livehoods are dynamic, and evolve as people's behaviours change, unlike the fixed neighbourhood borders set by the city government.
- **Demographics:** the interviews displayed strong evidence that the demographics of the residents and visitors of an area often play a strong role in explaining the divisions between livehoods.
- **Development and Resources:** economic development can affect the character of an area. Similarly, the resources (or lack there of) provided by a region has a strong influence on the people that visit it, and hence its resulting character. This is assumed to be reflected in the livehoods.
- **Geography and Architecture:** the movements of people through a certain area is presumably shaped by its geography and architecture; livehoods can reveal this influence and the effects it has over visiting patterns.

Take-Away k—means is not the sole clustering algorithm in applications!

6. Issues and Challenges

The Stench of Bad Data

We all say we like data, but we don't. We like getting insight out of data. That's not quite the same as liking data itself. In fact, I dare say that I don't quite care for data, and it sounds like I'm not alone.

– Q.E. McCallum, *Bad Data Handbook*

The data science landscape is littered with issues and challenges. We shall briefly discuss some of these in this section.

6.1 Bad Data

The main difficulties with data is that it is not always **representative** of the situation that we would like to model and that it might not be **consistent** (the collection and collation methods may have changed over time, say).

There are other potential data issues [34]:

- the data might be formatted for human consumption, not machine readability;
- the data might contain lies and mistakes;
- the data might not reflect reality, and
- there might be additional sources of bias and errors (not only imputation bias, but replacing extreme values with average values, proxy reporting, etc.).

Seeking perfection in the data beyond a "reasonable" threshold[17] can hamper the efforts of analysts: different quality requirements exist for academic data, professional data, economic data, government data, military data, service data, commercial data, etc.

It can be helpful to remember the engineering dictum: "close enough is good enough" (in terms of completeness, coherence, correctness, and accountability). The challenge lies in defining what is "close enough" for the application under consideration.

Even when all (most?) data issues have been mitigated, there remains a number of common **data analysis pitfalls**:

- analyzing data **without understanding the context**;
- using **one and only one tool** (by choice or by fiat) – neither the cloud, nor Big Data, nor Deep Learning, nor Artificial Intelligence will solve all of an organization's problems;
- analyzing data just for the sake of analysis;
- having **unrealistic expectations** of data analysis/DS/ML/AI – in order to optimize the production of actionable insights from data, we must first recognize the methods' domains of application and their limitations.

[17]This threshold is difficult to establish exactly, however.

6.2 Overfitting

In a traditional statistical model, p—values and goodness-of-fit statistics are used to validate the model. But such statistics cannot always be computed for predictive data science models. We recognise a "good" model based on how well **it performs on unseen data**.

In practice, training sets and ML methods are used to search for **rules** and **models** that are **generalizable to new data** (or validation/testing sets).

Problems arise when knowledge that is gained from supervised learning does not generalize properly to the data. Ironically, this may occur if the rules or models **fit the training set too well** – in other words, the results are too specific to the training set (see Figure 30 for an illustration of **overfitting** and **underfitting**).

A simple example may elucidate further. Consider the following set of rules regarding hair colour among humans:

- **vague rule** – some people have black hair, some have brown hair, some blond, and some red (this is obviously "true", but too general to be useful for predictions);
- **reasonable rule** – in populations of European descent, approximately 45% have black hair, 45% brown hair, 7% blond and 3% red;
- **overly specific rule** – in every 10,000 individuals of European descent, we predict there are 46.32% with black hair, 47.27% with brown hair, 6.51% with blond hair, and 0.00% with red hair (this rule presumably emerges from redhead-free training data).

With the overly specific rule, we would predict that there are no redheads in populations of European descent, which is blatantly false. This rule is **too specific** to the particular training subset that was used to produce it.[18]

More formally, underfitting and overfitting can be viewed as resulting from the **level of model complexity** (see Figure 31).

Underfitting can be overcome by using more complex models (or models that use more of a dataset's variables). Overfitting, on the other hand, can be overcome in several ways:

- **using multiple training sets** (ensemble learning approaches), with overlap being allowed – this has the effect of reducing the odds of finding spurious patterns based on quirks of the training data;
- **using larger training sets** may also remove signal which is too specific to a small training set – a 70% - 30% split is often suggested, and
- **using simpler models** (or models that use a dataset with a reduced number of variables as input).

[18]We could argue that the data was simply not representative – using a training set with redheads would yield a rule that would make better predictions. But "over-reporting/overconfidence" (which manifest themselves with the use of significant digits) are also part of the problem.

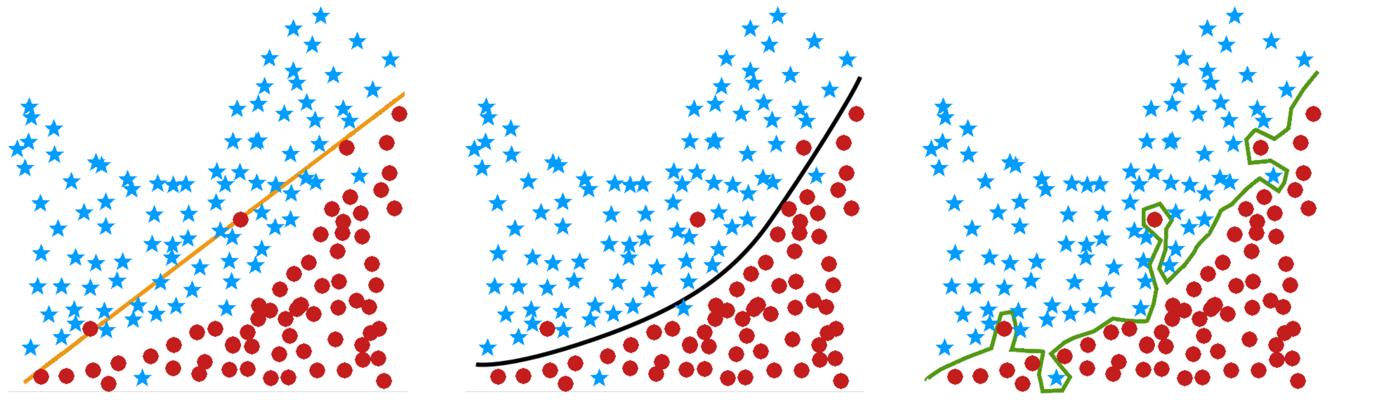


Figure 30. An illustration of underfitting (left) – the rule is not very accurate but easily generalizable to new instances – and overfitting (right) – very accurate but not easily generalizable to new instances. The middle fit is more reasonable.

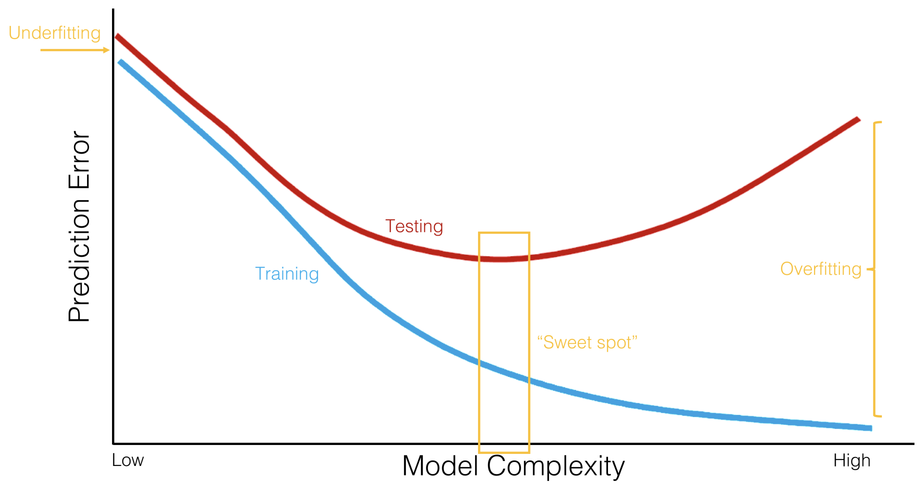


Figure 31. Underfitting and overfitting as a function of model complexity; error prediction on training sample (blue) and testing sample (red). High error prediction rates for simple models are a manifestation of underfitting; large difference between error prediction rates on training and testing samples for complex models are a manifestation of overfitting. Ideally, model complexity would be chosen to reach the situation's "sweet spot", but fishing for the ideal scenario might diminish explanatory power (based on [21]).

When using multiple training sets, the size of the dataset may also affect the suggested strategy: when faced with

- **small datasets** (less than a few hundred observations, say, but that depends on numerous factors such as computer power and number of tasks), use 100-200 repetitions of a **bootstrap procedure** [24];
- **average-sized datasets** (less than a few thousand observations), use a few repetitions of 10-fold cross-validation [24, 50] (see Figure 32);
- **large datasets**, use a few repetitions of a holdout split (70%-30%, say).

No matter which strategy is eventually selected, the machine learning approach requires ALL models to be evaluated on **unseen data**.

6.3 Appropriateness and Transferability

Data science models will continue to be used heavily in the near future; while there are pros and cons to their use on ethical and other non-technical grounds, their applicability is also driven by **technical considerations**.

DS/ML/AI methods are **not** appropriate if:

- existing (legacy) datasets absolutely must be used instead of ideal/appropriate datasets;[19]
- the dataset has attributes that usefully predict a value of interest, but these attributes **are not available** when a prediction is required (e.g. the total time spent on a website may be predictive of a visitor's future purchases, but the prediction must be made before the total time spent on the website is known);

[19]"It's the best data we have!" does not mean it's the right one.

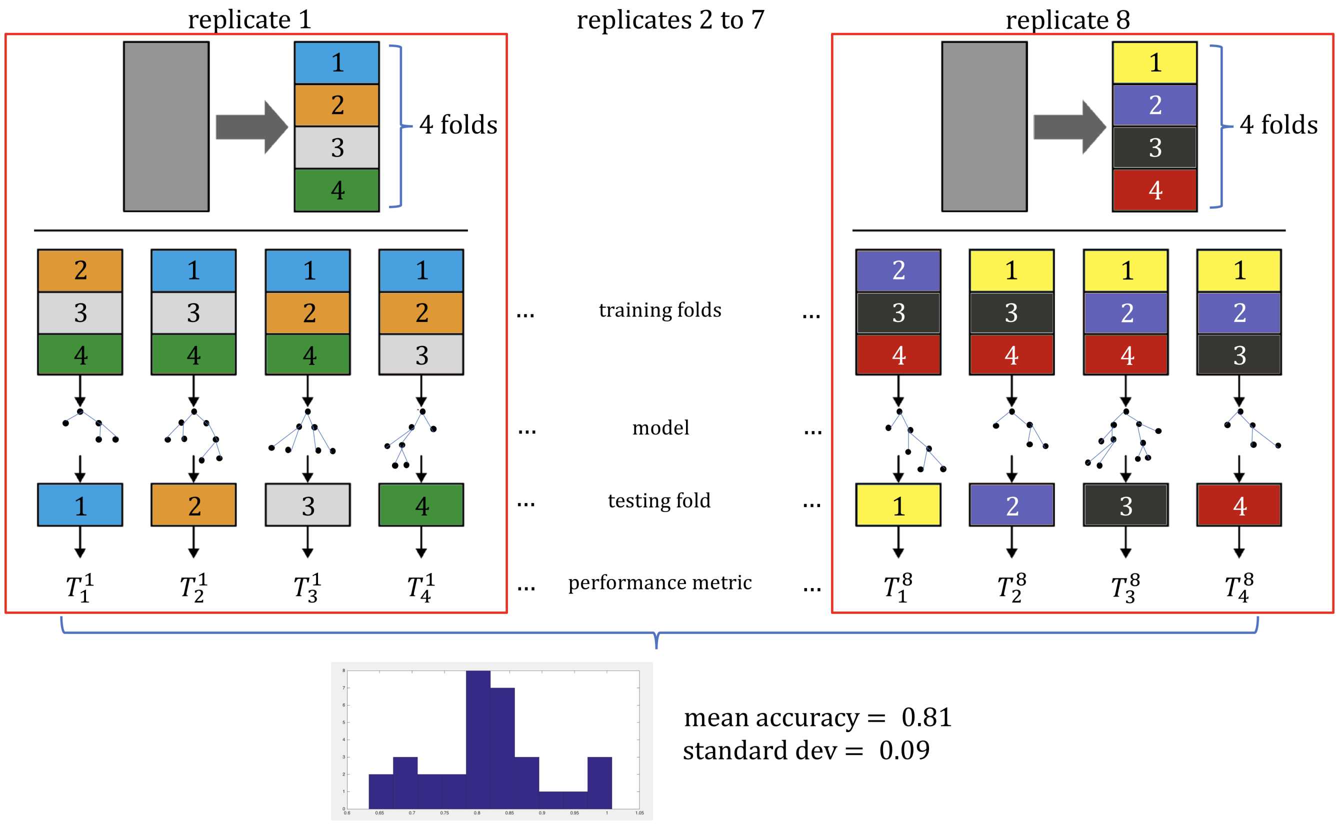


Figure 32. Schematic illustration of cross-fold validation, for 8 replicates and 4 folds; $8 \times 4 = 32$ models from a given family are built on various training sets (consisting of 3/4 of the available data – the training folds). Model family performance is evaluated on the respective holdout folds; the distribution of the performance metrics (in practice, some combination of the mean/median and standard deviation) can be used to compare various model families (based on [42, 50]).

- class membership or numerical outcome is going to be predicted using an unsupervised learning algorithm (e.g. clustering loan default data might lead to a cluster contains many defaulters – if new instances get added to this cluster, should they automatically be viewed as loan defaulters?).

Every model makes certain assumptions about what is and is not **relevant** to its workings, but there is a tendency to only gather data which is **assumed** to be relevant to a particular situation.

If the data is used in other contexts, or to make predictions depending on attributes for which no data is available, then there might be no way to **validate the results.**[20]

This is not as esoteric a consideration as it might seem: **over-generalizations and inaccurate predictions can lead to harmful results**.

[20]For instance, can we use a model that predicts whether a borrower will default on a mortgage or not to also predict whether a borrower will default on a car loan or not? The problem is compounded by the fact that there might be some link between mortgage defaults and car loan defaults, but the original model does not necessarily takes this into account.

6.4 Pitfalls and Mistakes

We end this chapter with lists (based on [33]) of DS **myths**:

1. DS is about algorithms
2. DS is about predictive accuracy
3. DS requires a data warehouse
4. DS requires a large quantity of data
5. DS requires only technical experts

and mistakes:

1. selecting the wrong problem
2. getting by without metadata understanding.
3. not planning the data analysis process
4. insufficient business/domain knowledge
5. using incompatible data analysis tools
6. using tools that are too specific
7. favouring aggregates over individual results
8. running out of time
9. measuring results differently than the client
10. naïvely believing what one is told about the data

It remains the consultant's responsibility to address these issues with the client, **the earlier, the better**. Do not assume that you are all on the same page – prod and ask.

References

[1] C. Aggarwal, editor. *Data Classification: Algorithms and Applications*. CRC Press, 2015.

[2] C. C. Aggarwal and C. K. Reddy, editors. *Data Clustering: Algorithms and Applications*. CRC Press, 2014.

[3] C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, page 18–24, New York, NY, USA, 1998. Association for Computing Machinery.

[4] F. R. Bach and M. I. Jordan. Learning spectral clustering, with application to speech separation. *J. Mach. Learn. Res.*, 7:1963–2001, Dec. 2006.

[5] S. E. Brossette, A. P. Sprague, J. M. Hardin, K. B. Waites, W. T. Jones, and S. A. Moser. Association Rules and Data Mining in Hospital Infection Control and Public Health Surveillance. *Journal of the American Medical Informatics Association*, 5(4):373–381, 07 1998.

[6] S. Canada. Athlete Rebate .

[7] J. Chambers and T. Hastie. *Statistical Models in S*. Wadsworth and Brooks/Cole, 1992.

[8] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In L. A. Adamic, R. Baeza-Yates, and S. Counts, editors, *ICWSM*. The AAAI Press, 2011.

[9] T. Chou. Apriori: Association Rule Mining In-depth Explanation and Python Implementation . *Towards Data Science*, Oct 2020.

[10] J. Cranshaw, R. Schwartz, J. I. Hong, and N. M. Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. In J. G. Breslin, N. B. Ellison, J. G. Shanahan, and Z. Tufekci, editors, *ICWSM*. The AAAI Press, 2012.

[11] T. Davenport and D. Patil. Data Scientist: The Sexiest Job of the 21st Century . *Harvard Business Review*, Oct 2012.

[12] B. Desgraupes. *clusterCrit: Clustering Indices*, 2018. R package version 1.2.8.

[13] J. d'Huy. Scientists trace society's myths to primordial origins. *Scientific American (Online)*, Sep 2016.

[14] D. Dua and E. Karra Taniskidou. UCI Machine Learning Repository , 2017.

[15] S. Fefilatyev, K. Kramer, L. Hall, D. Goldgof, R. Kasturi, A. Remsen, and K. Daly. Detection of anomalous particles from deepwater horizon oil spill using SIPPER3 underwater imaging platform. In *Data Mining Case Studies IV, Proceedings of the 11th IEEE International Conference on Data Mining*. IEEE, Vancouver, BC, 2011.

[16] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

[17] E. Garcia, C. Romero, S. Ventura, and T. Calders. Drawbacks and solutions of applying association rule mining in learning management systems. In *Proceedings of the International Workshop on Applying Data Mining in e-Learning*, Greece, 2007. CEUR-WS.org.

[18] U. Habib, K. Hayat, and G. Zucker. Complex building's energy system operation patterns analysis using bag of words representation with hierarchical clustering. *Complex Adapt. Syst. Model.*, 4:8, 2016.

[19] M. Hahsler and K. Hornik. New probabilistic interest measures for association rules. *CoRR*, abs/0803.0966, 2008.

[20] N. Harris. Visualizing DBSCAN Clustering .

[21] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed.* Springer, 2008.

[22] T. Hastie, T. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

[23] K.-W. Hsu, N. Pathak, J. Srivastava, G. Tschida, and E. Bjorklund. *Data Mining Based Tax Audit Selection: A Case Study of a Pilot Project at the Minnesota Department of Revenue*, pages 221–245. Springer International Publishing, Cham, 2015.

[24] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer, 2014.

[25] A. Jawad, K. Kersting, and N. Andrienko. Where traffic meets dna: Mobility mining using biological sequence analysis revisited. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, page 357–360, New York, NY, USA, 2011. Association for Computing Machinery.

[26] A. Jensen, P. Moseley, T. Oprea, S. Ellesøe, R. Eriksson, H. Schmock, P. Jensen, L. Jensen, and S. Brunak. Temporal disease trajectories condensed from population-wide registry data covering 6.2 million patients. *Nature Communications*, 5, 2014.

[27] B. Kitts. Product targeting from rare events: Five years of one-to-one marketing at CPI. *Marketing Science Conference*, 2005.

[28] B. Kitts. The making of a large-scale ad server. In *Data Mining Case Studies Workshop and Practice Prize 5*, Dallas, TX, 2013. Proceedings of the IEEE Thirteenth International Conference on Data Mining Workshops, IEEE Press.

[29] B. Kitts, J. Zhang, G. Wu, W. Brandi, J. Beasley, K. Morrill, J. Ettedgui1, S. Siddhartha, H. Yuan, F. Gao, P. Azo, and R. Mahato. Click fraud detection: Adversarial pattern recognition over 5 years at microsoft. In *Annals of Information Systems (Special Issue on Data Mining*

in Real-World Applications), pages 181–201. Springer, 2015.

[30] H. T. Kung and D. Vlah. A spectral clustering approach to validating sensors via their peers in distributed sensor networks. *Int. J. Sen. Netw.*, 8(3/4):202–208, Oct. 2010.

[31] O. Leduc and P. Boily. Boosting with AdaBoost and Gradient Boosting. *Data Action Lab Blog*, 2019.

[32] J. Leskovec, A. Rajamaran, and J. Ullman. *Mining of Massive Datasets*. Cambridge Press, 2014.

[33] A. Maheshwari. *Business Intelligence and Data Mining*. Business Expert Press, 2015.

[34] Q. McCallum. *Bad Data Handbook*. O'Reilly, 2013.

[35] A. Ng and K. Soo, editors. *Surviving a Disaster, in Numsense!* algobeans, 2016.

[36] E. Omiecinski. Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):57–69, 2003.

[37] M. Orlowska, K. Pytlakowska, A. Mrozek-Wilczkiewicz, R. Musiol, M. Waksmundzka-Hajnos, M. Sajewicz, and T. Kowalska. A comparison of antioxidant, antibacterial, and anticancer activity of the selected thyme species by means of hierarchical clustering and principal component analysis. *Acta Chromatographica Acta Chromatographica*, 28(2):207 – 221, 2016.

[38] V. U. Panchami and N. Radhika. A novel approach for predicting the length of hospital stay with dbscan and supervised classification algorithms. In *ICADIWT*, pages 207–212. IEEE, 2014.

[39] V. U. Panchami and N. Radhika. A novel approach for predicting the length of hospital stay with dbscan and supervised classification algorithms. In *ICADIWT*, pages 207–212. IEEE, 2014.

[40] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, 1991.

[41] C. Plant, S. J. Teipel, A. Oswald, C. Böhm, T. Meindl, J. M. Miranda, A. L. W. Bokde, H. Hampel, and M. Ewers. Automated detection of brain atrophy patterns based on mri for the prediction of alzheimer's disease. *NeuroImage*, 50(1):162–174, 2010.

[42] F. Provost and T. Fawcett. *Data Science for Business*. O'Reilly, 2015.

[43] A. M. Raja. Penguins Dataset Overview - iris alternative. *Towards Data Science*, Jun 2020.

[44] M. Risdal. Exploring survival on the titanic. *Kaggle.com*, 2016.

[45] D. Robinson. What's the difference between data science, machine learning, and artificial intelligence? *Variance Explained*, Jan 2018.

[46] G. Schoier and G. Borruso. Individual movements and geographical data mining. clustering algorithms for highlighting hotspots in personal navigation routes. In B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, and B. O. Apduhan, editors, *Computational Science and Its Applications - ICCSA 2011*, pages 454–465, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[47] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, 42(3), July 2017.

[48] E. Siegel. *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie or Die*. Predictive Analytics World, 2016.

[49] P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right objective measure for association analysis. *Inf. Syst.*, 29(4):293–313, June 2004.

[50] L. Torgo. *Data Mining with R, 2nd ed.* CRC Press, 2016.

[51] Wikipedia. Cluster Analysis Algorithms.

[52] Wikipedia. Association Rule Learning, 2020.

[53] D. Wolpert and W. Macready. Coevolutionary free lunches. *IEEE Transactions on Evolutionary Computation*, 9(6):721–735, 2005.

[54] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[55] D. Woods. bitly's Hilary Mason on "What is A Data Scientist?". *Forbes*, Mar 2012.