Texts in Computer Science

Michael R. Berthold · Christian Borgelt
Frank Höppner · Frank Klawonn
Rosaria Silipo

Guide to Intelligent Data Science

How to Intelligently Make Use of Real Data

Second Edition

Springer

Michael R. Berthold
Department of Computer and Information Science
University of Konstanz
Konstanz, Germany

Christian Borgelt
Department of Computer Sciences
University of Salzburg
Salzburg, Austria

Series Editors
David Gries
Department of Computer Science
Cornell University
Ithaca, NY, USA

Frank Höppner
Department of Computer Science
Ostfalia University of Applied Sciences
Wolfenbüttel, Germany

Frank Klawonn
Department of Computer Science
Ostfalia University of Applied Sciences
Wolfenbüttel, Germany

Rosaria Silipo
KNIME AG
Zurich, Switzerland

Orit Hazzan
Faculty of Education in Technology and Science
Technion – Israel Institute of Technology
Haifa, Israel

ISSN 1868-0941
Texts in Computer Science
ISBN 978-3-030-45573-6
https://doi.org/10.1007/978-3-030-45574-3

ISSN 1868-095X (electronic)

ISBN 978-3-030-45574-3 (eBook)

Contents

Symbols

A, A_i	attribute, variable [e.g., $A_1 = color$, $A_2 = price$, $A_3 = category$]
ω	a possible value of an attribute [e.g., $\omega = red$]
Ω, dom$(\cdot)$	set of possible values of an attribute [e.g., $\Omega_1 = \Omega_{\text{color}} = \text{dom}(A_i) = \{red, blue, green\}$]
$\mathcal{A}$	set of all attributes [e.g., $\mathcal{A} = \{color, price, category\}$]
m	number of considered attributes [e.g., 3]
x	a specific value of an attribute [e.g., $x_2 = x_{\text{price}} = 4000$]
$\mathcal{X}$	space of possible data records [e.g., $\mathcal{X} = \Omega_{A_1} \times \cdots \times \Omega_{A_m}$]
$\mathcal{D}$	set of all records, data set, $\mathcal{D} \subseteq \mathcal{X}$ [e.g., $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$]
n	number of records in data set
$\mathbf{x}$	record in database [e.g., $\mathbf{x} = (x_1, x_2, x_3) = (red, 4000, luxury)$]
$\mathbf{x}_A$	attribute A of record $\mathbf{x}$ [e.g., $\mathbf{x}_{\text{price}} = 4000$]
$\mathbf{x}_{2,A}$	attribute A of record $\mathbf{x}_2$
$\mathcal{D}_{A=v}$	set of all records $\mathbf{x} \in \mathcal{D}$ with $\mathbf{x}_A = v$
C	a selected categorical target attribute [e.g., $C = A_3 = category$]
Ω_C	set of all possible classes [e.g., $\Omega_C = \{\text{quits, stays, unknown}\}$]
Y	a selected continuous target attribute [e.g., $Y = A_2 = price$]
$\mathcal{C}$	cluster (set of associated data objects) [e.g., $\mathcal{C} \subseteq \mathcal{D}$]
c	number of clusters
$\mathcal{P}$	partition, set of clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_c\}$
$p_{i\|j}$	membership degree of data #j to cluster #i
$[p_{i\|j}]$	membership matrix
$d_\cdot$	distance function, metric (d_E: Euclidean)
$[d_{i,j}]$	distance matrix

Introduction 1

In this introductory chapter we provide a brief overview of some core ideas of data science and their motivation. In a first step we carefully distinguish between "data" and "knowledge" in order to obtain clear notions that help us to work out why it is usually not enough to simply collect data and why we have to strive to turn them into knowledge. As an illustration, we consider a well-known example from the history of science. In a second step we characterize the data science process, also often referred to as the knowledge discovery process, in which modeling is one important step. We characterize standard data science tasks and summarize the catalog of methods to tackle them.

1.1 Motivation

Every year that passes brings us more powerful computers, faster and cheaper storage media, and higher bandwidth data connections. Due to these technological advancements, it is possible nowadays to collect and store enormous amounts of data with little effort and at impressively low costs. As a consequence, more and more companies, research centers, and governmental institutions create huge archives of tables, documents, images, and sounds in electronic form. Since for centuries lack of data has been a core hindrance to scientific and economic progress, we feel compelled to think that we can solve—at least in principle—basically any problem we are faced with if only we have enough data.

However, a closer examination of the matter reveals that this is an illusion. Data alone, regardless of how voluminous they are, are not enough. Even though large databases allow us to retrieve many different single pieces of information and to compute (simple) aggregations (like average monthly sales in Berlin), general patterns, structures, and regularities often go undetected. We may say that in the vast amount of data stored in some data repositories we cannot see the wood (the patterns) for the trees (the individual data records). However, it is most often exactly these patterns, regularities, and trends that are particularly valuable if one desires, for example, to increase sales in a supermarket. Suppose, for instance, a supermarket manager discovers that by analyzing sales and customer records certain products

M.R. Berthold et al., *Guide to Intelligent Data Science*,
https://doi.org/10.1007/978-3-030-45574-3_1

are frequently bought together. In such a case sales can sometimes be stimulated by cleverly arranging these products on the shelves of the market (they may, for example, be placed close to each other, or may be offered as a bundle, in order to invite even more customers to buy them together).

Unfortunately, it turns out to be harder than may be expected at first sight to actually discover such patterns and regularities and thus to exploit a larger part of the information that is contained in the available data. In contrast to the overwhelming flood of data there was, at least at the beginning, a lack of tools by which raw data could be transformed into useful information. More than 20 years ago John Naisbett aptly characterized the situation by saying [4]: "We are drowning in information, but starving for knowledge." As a consequence, a new research area has been developed, which has become known under the name of *data science*. The goal of this area was to meet the challenge to develop tools that can help humans to find potentially useful patterns in their data and to solve the problems they are facing by making better use of the data they have. Today, more than 20 years later, a lot of progress has been made, and a considerable number of methods and implementations of these techniques in software tools have been developed. Still it is not the tools alone, but the *intelligent composition* of human intuition with the computational power, of sound background knowledge with computer-aided modeling, of critical reflection with convenient automatic model construction, that leads *data science* projects to success [2]. In this book we try to provide a hands-on approach to many basic data science techniques and how they are used to solve data science problems if relevant data is available.

1.1.1 Data and Knowledge

In this book we distinguish carefully between *data* and *knowledge*. Statements like "Columbus discovered America in 1492" or "Mister Smith owns a VW Beetle" are **data**. Note that we ignore whether we already know these statements or whether we have any concrete use for them at the moment. The essential property of these statements we focus on here is that they refer to single events, objects, people, points in time, etc. That is, they generally refer to single instances or individual cases. As a consequence, their domain of application and thus their utility is necessarily limited.

In contrast to this, **knowledge** consists of statements like "All masses attract each other" or "Every day at 7:30 a.m. a train with destination Rome departs from Zurich main station." Again, we neglect the relevance of these statements for our current situation and whether we already know them. Rather, we focus on the essential property that they do *not* refer to single instances or individual cases but are general rules or (physical) laws. Hence, if they are true, they have a large domain of application. Even more importantly, though, they allow us to make predictions and are thus highly useful (at least if they are relevant to us).

We have to admit, though, that in daily life we also call statements like "Columbus discovered America in 1492" knowledge (actually, this particular statement is used as a kind of prototypical example of knowledge). However, we neglect here

this vernacular and rather fuzzy use of the notion "knowledge" and express our regrets that it is not possible to find a terminology that is completely consistent with everyday speech. Neither single statements about individual cases nor collections of such statements qualify, in our use of the term, as knowledge.

Summarizing, we can characterize data and knowledge as follows:

data

- refer to single instances
  (single objects, people, events, points in time, etc.)
- describe individual properties
- are often available in large amounts
  (databases, archives)
- are often easy to collect or to obtain
  (e.g., scanner cashiers in supermarkets, Internet)
- do not allow us to make predictions or forecasts

knowledge

- refers to *classes* of instances
  (*sets* of objects, people, events, points in time, etc.)
- describes general patterns, structures, laws, principles, etc.
- consists of as few statements as possible
  (this is actually an explicit goal, see below)
- is often difficult and time consuming to find or to obtain
  (e.g., natural laws, education)
- allows us to make predictions and forecasts

These characterizations make it very clear that generally knowledge is much more valuable than (raw) data. Its generality and the possibility to make predictions about the properties of new cases are the main reasons for this superiority.

It is obvious, though, that not all kinds of knowledge are equally valuable as any other. Not all general statements are equally important, equally substantial, equally significant, or equally useful. Therefore knowledge has to be assessed, so that we do not drown in a sea of irrelevant knowledge. The following list (which we do not claim to be complete) lists some of the most important criteria:

criteria to assess knowledge

- correctness (probability, success in tests)
- generality (domain and conditions of validity)
- usefulness (relevance, predictive power)
- comprehensibility (simplicity, clarity, parsimony)
- novelty (previously unknown, unexpected)

In the domain of science, the focus is on correctness, generality, and simplicity (parsimony) are in the focus: one way of characterizing science is to say that it is the search for a minimal, correct description of the world. In economy and industry,

however, the emphasis is placed on usefulness, comprehensibility, and novelty: The main goal is to gain a competitive edge and thus to increase revenues. Nevertheless, neither of the two areas can afford to neglect the other criteria.

1.1.2 Tycho Brahe and Johannes Kepler

We illustrate the considerations of the previous section with an (at least partially) well-known example from the history of science. In the sixteenth century studying the stars and the planetary motions was one of the core areas of research. Among its proponents was Tycho Brahe (1546–1601), a Danish nobleman and astronomer, who in 1576 and 1584, with the financial help of King Frederic II, built two observatories on the island of Ven, about 32 km north-east of Copenhagen. He had access to the best astronomical instruments of his time (but no telescopes, which were used only later by Galileo Galilei (1564–1642) and Johannes Kepler (see below) to observe celestial bodies), and he used them to determine the positions of the sun, the moon, and the planets with a precision of less than one angle minute. With this precision he managed to surpass all measurements that had been carried out before and to actually reach the theoretical limit for observations with the unaided eye (that is, without the help of telescopes). Working carefully and persistently, he recorded the motions of the celestial bodies over several years.

Stated plainly, Tycho Brahe collected data about our planetary system, fairly large amounts of data, at least from the point of view of the sixteenth century. However, he failed to find a consistent scheme to combine them, could not discern a clear underlying pattern—partially because he stuck too closely to the geocentric system (the earth is in the center, and all planets, the sun, and the moon revolve around the earth). He could tell the precise location of Mars on any given day of the year 1582, but he could not connect its locations on different days by a clear and consistent theory. All hypotheses he tried did not fit his highly precise data. For example, he developed the so-called Tychonic planetary system (the earth is in the center, the sun and the moon revolve around the earth, and the other planets revolve around the sun on circular orbits). Although temporarily popular in the seventeenth century, this system did not stand the test of time. From a modern point of view, we may say that Tycho Brahe had a "data science problem" (or "knowledge discovery problem"). He had obtained the necessary data but could not extract the hidden knowledge.

This problem was solved later by Johannes Kepler (1571–1630), a German astronomer and mathematician, who worked as an assistant of Tycho Brahe. Contrary to Brahe, he advocated the Copernican planetary system (the sun is in the center, the earth and all other planets revolve around the sun in circular orbits) and tried all his life to reveal the laws that govern the motions of the celestial bodies. His approach was almost radical for his time, because he strove to find a mathematical description. He started his investigations with the data Tycho Brahe had collected and which he extended in later years. After several fruitless trials and searches and long and cumbersome calculations (imagine no pocket calculators), Kepler finally succeeded. He managed to combine Tycho Brahe's data into three simple laws, which nowadays bear his name as **Kepler's laws**. After having realized in 1604 already that

the course of Mars is an ellipse, he published the first two of these laws in his work "Astronomia Nova" in 1609 [7] and the third law ten years later in his magnum opus "Harmonices Mundi" [5, 8]:

1. The orbit of every planet (including the earth) is an ellipse, with the sun at a focal point.
2. A line from the sun to the planet sweeps out equal areas during equal intervals of time.
3. The squares of the orbital periods of any two planets relate to each other like the cubes of the semimajor axes of their respective orbits: $T_1^2/T_2^2 = a_1^3/a_2^3$, and therefore generally $T \sim a^{\frac{3}{2}}$.

Tycho Brahe had collected a large amount of astronomical data, and Johannes Kepler found the underlying laws that can explain them. He discovered the hidden knowledge and thus became one of the first "data scientists" in history.

Today the works of Tycho Brahe are almost forgotten—few have even heard his name. His catalogs of celestial data are merely of historical interest. No textbook on astronomy contains excerpts from his measurements—and this is only partially due to the better measurement technology we have available today. His observations and precise measurements are raw data and thus suffer from a decisive drawback: they do not provide any insight into the underlying mechanisms and thus do not allow us to make predictions. Kepler's laws, on the other hand, are treated in basically all astronomy and physics textbooks, because they state the principles according to which planets and comets move. They combine all of Brahe's observations and measurements in three simple statements. In addition, they permit us to make predictions: if we know the location and the speed of a planet relative to the sun at any given moment, we can compute its future course by drawing on Kepler's laws.

How did Johannes Kepler find the simple astronomical laws that bear his name? How did he discover them in Tycho Brahe's long tables and voluminous catalogs, thus revolutionizing astronomy? We know fairly little about his searches and efforts. He must have tried a large number of hypotheses, most of them failing. He must have carried out long and cumbersome computations, repeating some of them several times to eliminate errors. It is likely that exceptional mathematical talent, hard and tenacious work, and a significant amount of good luck finally led him to success. What we can be sure of is that he did not possess a universally applicable procedure or method to discover physical or astronomical laws.

Even today we are not much further: there is still no silver bullet to hit on the right solution. It is still much easier to collect data, with which we are virtually swamped in today's "information society" (whatever this popular term actually means) than to discover knowledge. Automatic measurement instruments and scanners, digital cameras and computers, and an abundance of other automatic and semiautomatic devices have even relieved us of the burden of manual data collection. In addition, modern databases and the ability to store data in the cloud allow us to keep ever increasing amounts of data and to retrieve and to sample them easily. John Naisbett was perfectly right: "We are drowning in information, but starving for knowledge."

It took a distinguished researcher like Johannes Kepler several years (actually half a lifetime) to evaluate the data that Tycho Brahe had collected—data that from a modern point of view are negligibly few and of which Kepler actually analyzed closely only those about the orbit of Mars. Given this, how can we hope today to cope with the enormous amounts of data we are faced with every day? "Manual" analyses (like Kepler's) have long ceased to be feasible. Simple aids, like the visualization of data in charts and diagrams, even though highly useful and certainly a first and important step, quickly reach their limits. Thus, if we refuse to surrender to the flood of data, we are forced to develop and employ computer-aided techniques, with which data science can be simplified or even automated to some degree. These are the methods that have been and still are developed in the research areas of statistics, machine learning, data analysis, knowledge discovery in databases, and data mining. Even though these methods are far from replacing human beings like Johannes Kepler, especially since a mindless (or unintelligent) application can easily produce artifacts and misleading results, it is not entirely implausible to assume that Kepler, if he had been supported by these methods and tools, could have reached his goal a little earlier.

1.1.3 Intelligent Data Science

Many people associate any kind of data science with **statistics** (see also Appendix A, which provides a brief review). Statistics has a long history and originated from collecting and analyzing data about the population and the state in general.

Statistics can be divided into *descriptive* and *inferential statistics*. **Descriptive statistics** summarizes data without making specific assumptions about the data, often by characteristic values like the (empirical) mean or by diagrams like histograms. **Inferential statistics** provides more rigorous methods than descriptive statistics that are based on certain assumptions about the data generating random process. The conclusions drawn in inferential statistics are only valid if these assumptions are satisfied.

Typically, in statistics the first step of the data science process is to *design the experiment* that defines how data should be collected in order to be able to carry out a reliable analysis based on the obtained data. To capture this important issue, we distinguish between *experimental* and *observational studies*. In an **experimental study** one can control and manipulate the data generating process. For instance, if we are interested in the effects of certain diets on the health status of a person, we might ask different groups of people to stick to different diets. Thus we have a certain control over the data generating process. In this experimental study, we can decide which and how many people should be assigned to a certain diet.

In an **observational study** one cannot control the data generating process. For the same dietary study as above, we might simply ask people on the street what they normally eat. Then we have no control about which kinds of diets we get data and how many people we will have for each diet in our data.

No matter whether the study is experimental or observational, there are usually independence assumptions involved, and the data we collect should be representative. The main reason is that inferential statistics is often applied to *hypothesis testing* where, based on the collected data, we desire to either confirm or reject some hypothesis about the considered domain. In this case representative data and certain independencies are required in order to ensure that the test decisions are valid.

In contrast to hypothesis testing, **exploratory data analysis** is concerned with *generating hypotheses* from the collected data. In exploratory data analysis there are no or at least considerably weaker model assumptions about the data generating process. Most of the methods presented in this book fall into this category, since they are mostly universal methods designed to achieve a certain goal but are not based on a rigorous model as in inferential statistics.

The typical situation we assume in this book is that we already have the data. They might not have been collected in the best way, or in the way we would have collected them had we been able to design the experiment in advance. Therefore, it is often difficult to make specific assumptions about the data generating process. We are also mostly goal-oriented—that is, we ask questions like "Which customers will yield the highest profit"?—and search for methods that can help us to answer such questions or to solve our problems.

The opportunity of analyzing large real world data repositories that were initially collected for different purposes came with the availability of powerful tools and technologies that can process and analyze massive amounts of data, which is nowadays called **data science**.

In this book we strove to provide a comprehensive guide to intelligent data science, outlining the process and its phases, presenting methods and algorithms for various tasks and purposes, and illustrating them using a well known, freely available software platform. In this way we hope to offer a good starting point for anyone who wishes to become more familiar with the area of data science.

1.2 The Data Science Process

There are at least two typical situations in which data science may help us to find solutions to certain problems or provide answers to questions that arise. In the first case, the problem at hand is by no means new, but it is already solved as a matter of routine (e.g., approval of credit card applications, technical inspection during quality assurance, machine control by a plant operator, etc.). If data have been collected for the past cases together with the result that was finally achieved (such as poor customer performance, malfunction of parts, etc.), such historical data may be used to revise and optimize the presently used strategy to reach a decision. In the second case, a certain question arises for the first time, and only little experience is available, or the experience is not directly applicable to this new question (e.g., starting with a new product, preventing abuse of servers, evaluating a large experiment or survey). In such cases, it is supposed that data from related situations may be helpful to generalize the new problem or that unknown relationships can be discovered from the data to gain insights into this unfamiliar area.

What if we have no data at all? This situation does not usually occur in practice, since in most cases there is always *some* data. Especially in businesses huge amounts of data have been collected and stored for operational reasons in the past (e.g., billing, logistics, warranty claims) that may now be used to optimize various decisions or offer new options (e.g., predicting customer performance, reducing stock on hand, tracking causes of defects). So the right question should be: How do we know if we have enough *relevant* data? This question is not answered easily. If it actually turns out that the data are not sufficient, one option is to acquire new data to solve the problem. However, as already pointed out in the preceding section, the experimental design of data acquisition is beyond the scope of this book.

Historically there have been several proposals about what the data science process should look like, such as SEMMA (an acronym for *sample, explore, modify, model, assess*), CRISP-DM (an acronym for *CRoss Industry Standard Process for Data Mining* as defined by the CRISP-DM consortium) [3], or the KDD-process [4] (see [9] for a detailed comparison). In this book, we are going to adopt the CRISP-DM process, which has been developed by a consortium of large companies, and still represents the most widely used process model for data science today.

Thus, the data science process consists of six phases as shown in Fig. 1.1. Most of these phases are usually executed more than once, and the most frequent phase transitions are shown by arrows. The main objective of the first **project understanding** step (see Chap. 3) is to identify the potential benefit, as well as the risks and efforts of a successful project, such that a deliberate decision on conducting the full project can be made. The envisaged solution is also transferred from the project domain to a more technical, data-centered notion. This first phase is usually called business understanding, but we stick to the more general term *project understanding* to emphasize that our problem at hand may as well be purely technical in nature or a research project rather than economically motivated.

Next we need to make sure that we will have sufficient data at hand to tackle the problem. While we cannot know this for sure until the end of the project, we at least have to convince ourselves that there are enough relevant data. To achieve this, we proceed in the **data understanding** phase (see Chap. 4) with a review of the available databases and the information contained in the database fields, a visual assessment of the basic relationships between attributes, a data quality audit, an inspection of abnormal cases (outliers), etc. For instance, outliers appear to be abnormal in some sense and are often caused by faulty insertion, but sometimes they give surprising insights on closer inspection. Some techniques respond very sensitively to outliers, which is why they should be treated with special care. Another aspect is empty fields which may occur in the data for various reasons—ignoring them may introduce a systematic error in the results. By getting familiar with the data, typically first insights and hypotheses are gained. If we do not believe that the data suffice to solve the problem, it may be necessary to revise the project's objective.

So far, we have not changed any field of our data. However, this will be required to get the data into a shape that enables us to apply modeling tools. In the **data preparation** phase (Chap. 6) the data are selected, corrected, modified, even new attributes are generated, such that the prepared data sets best suit the problem and the

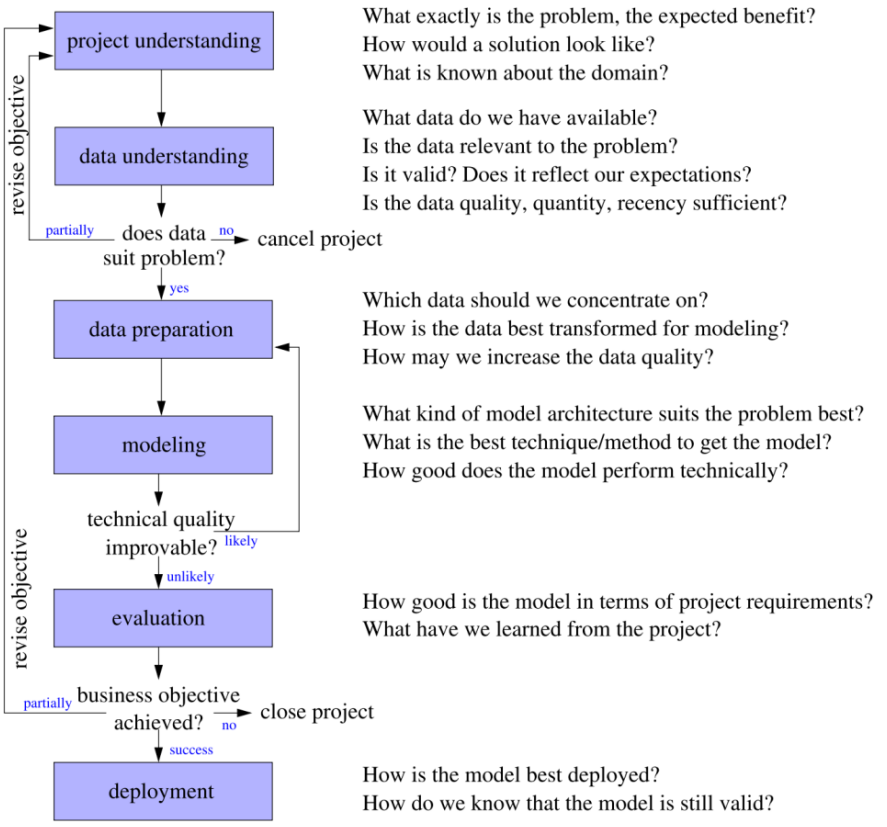

Fig. 1.1 Overview of the data science process together with typical questions to be asked in the respective phases

envisaged modeling technique. Basically all deficiencies that have been identified in the data understanding phase require special actions. Often the outliers and missing values are replaced by estimated values or true values obtained from other sources. We may restrict further analysis to certain variables and to a selection of the records from the full data set. Redundant and irrelevant data can give many techniques an unnecessarily hard time.

Once the data are prepared, we select and apply **modeling** tools to extract *knowledge* out of the data in the form of a *model* (Chaps. 5 and 7–9). Depending on what we want to do with the model, we may choose techniques that are easily interpretable (to gain insights) or less demonstrative black-box models, which may perform better. If we are not pleased with the results but are confident that the model can be improved, we step back to the data preparation phase and, say, generate new attributes from the existing ones, to support the modeling technique or to apply different techniques. Background knowledge may provide hints on useful *transformations* that simplify the representation of the solution.

Compared to the modeling itself, which is typically supported by efficient tools and algorithms, the data understanding and preparation phases take considerable part of the overall project time as they require a close manual inspection of the data, investigations into the relationships between different data sources, often even the analysis of the process that generated the data. New insights promote new ideas for feature generation or alter the subset of selected data, in which case the data preparation and modeling phases are carried out multiple times. The number of steps is not predetermined but influenced by the process and findings itself.

When the technical benchmarks cannot be improved anymore, the obtained results are analyzed in the **deployment** phase (Chap. 10) from the perspective of the problem owner. At this point, the project may stop due to unsatisfactory results, the objectives may be revised in order to succeed under a slightly different setting, or the found and optimized model may be deployed.

After **deployment**, which ranges from writing a report to the creation of a software system that applies the model automatically to aid or make decisions, the project is not necessarily finished. If the project results are used continuously over time, an additional monitoring and model management phase is necessary: During the analysis, a number of assumptions will be made, and the correctness of the derived model (and the decisions that rely on the model) depends on them. So we better verify from time to time that these assumptions still hold to prevent decision-making on outdated information.

In the literature one can find attempts to create cost models that estimate the costs associated with a data science project. Without going into the details, the major key factors that remained in a reduced cost model derived from 40 projects were [10]:

- the number of tables and attributes,
- the dispersion of the attributes (only a few vs. many values),
- the number of external data sources,
- the type of the model (prediction being the most expensive),
- the attribute type mixture (mixture of numeric and nonnumeric), and
- the familiarity of the staff with data science projects in general, the project domain in particular, and the software suites.

While there is not much we can do about the problem size, the goal of this book is to increase the familiarity with data science projects by going through each of the phases and providing first instructions to get along with the software suites.

It is obvious that this process requires lots of familiarity with the methods used, as well as experience with data science projects in general. So in recent years the question has arisen as to how less trained people can use data science techniques and how parts of or the entire process can be automated. Obviously, in order to apply sophisticated techniques, some understanding of the underlying theory is required—otherwise the results are potentially nonsensical. This is why we added the term "intelligent" in the title of this book. Our aim is to introduce sufficient details about the main steps of the data science process so that they can be applied in a thoughtful, intelligent way [6].

To address this, techniques for automated machine learning have been suggested that are able to automatically create, train, and test machine learning models with as little expert input as possible. There are different algorithms and strategies to do this which vary by complexity and performance, but the main idea is empowering business analysts to train a great number of models and deliver the best one with just a small amount of configuration [1, 11, 12]. These methods can work surprisingly well if the data are in appropriate shape but face limitations for less typical or more complex tasks. Systems for automation of other steps of the data science process appear increasingly, but they face similar limitations. More complex data science problems are likely to call for a higher degree of data science expertise in conjunction with business or problem insights. For example, the domain expert can add some unique knowledge about the data treatment and filtering before continuing with the machine learning process. Also, when the data domain becomes more complex than simple tabular data, e.g., includes text, images or time series, the human expert can contribute custom techniques for data preparation, data partitioning, and feature engineering.

The focus of this book is not on techniques to eliminate the need for data science expertise, but instead on providing deeper insights into the methods so that they can be applied in an intelligent way. This should not eliminate the use of some automation techniques where appropriate—but that decision alone already requires some level of data science expertise.

1.3 Methods, Tasks, and Tools

Problem Categories Every data science problem is different. To avoid the effort of inventing a completely new solution for each problem, it is helpful to think of different problem categories and consider them as building blocks from which a solution may be composed. These categories also help categorize the large number of different algorithms that solve specific tasks. Over the years, the following set of method categories has been established [4]:

- **Classification**
  Predict the outcome of an experiment with a finite number of possible results (like *yes*/*no* or *unacceptable*/*acceptable*/*good*/*very good*). We may be interested in a prediction because the true result will emerge in the future or because it is expensive, difficult, or cumbersome to determine it.
  Typical questions: *Is this customer credit-worthy? Will this customer respond to our mailing? Will the technical quality be acceptable?*
- **Regression**
  Regression is, just like classification, also a prediction task, but this time the value of interest is numerical in nature.
  Typical questions: *How will the EUR/USD exchange rate develop? How much money will the customer spend for vacation next year? How much will the machine's temperature change within the next cycle?*

- **Clustering, segmentation**
  Summarize the data to get a better overview by forming groups of similar cases (called clusters or segments). Instead of examining a large number of similar records, we need to inspect the group summary only. We may also obtain some insight into the structure of the whole data set. Cases that do not belong to any group may be considered as abnormal or outliers.
  Typical questions: *Do my customers divide into different groups? How many operating points does the machine have, and what do they look like?*
- **Association analysis**
  Find any correlations or associations to better understand or describe the interdependencies of all the attributes. The focus is on *relationships* between all attributes rather than on a single target variable or the cases (full record).
  Typical questions: *Which optional equipment of a car often goes together? How do the various qualities influence each other?*
- **Deviation analysis**
  Knowing already the major trends or structures, find any exceptional subgroup that behaves differently with respect to some target attribute.
  Typical questions: *Under which circumstances does the system behave differently? Which properties do those customers share who do not follow the crowd?*

The most frequent categories are *classification* and *regression*, because decision making always becomes much easier if reliable predictions of the near future are available. When a completely new area or domain is explored, cluster analysis and association analysis may help identify relationships among attributes or records. Once the major relationships are understood (e.g., by a domain expert), a deviation analysis can help focus on *exceptional situations* that deviate from regularity.

Catalog of Methods There are various methods in each of these categories to find reliable answers to the questions raised above. However, there is no such thing as a *single golden method* that works perfectly for all problems. To convey some idea which method may be best suited for a given problem, we will discuss various methods in Chaps. 7–9. However, in order to organize these chapters, we did not rely on the problem categories collected above, as some methods can be used likewise for more than one problem type. We rather used the intended task of the data analysis as a grouping criterion:

- **Finding patterns** (Chap. 7)
  If the domain (and therefore the data) is new to us or if we expect to find interesting relationships, we explore the data for new, previously unknown patterns. We want to get a full picture and do not concentrate on a single target attribute, yet. We may apply methods from, for instance, segmentation, clustering, association analysis, or deviation analysis.
- **Finding explanations** (Chap. 8)
  We have a special interest in some target variable and wonder why and how it varies from case to case. The primary goal is to gain new insights (knowledge) that may influence our decision making, but we do not necessarily intend au-

tomation. We may apply methods from, for instance, classification, regression, association analysis, or deviation analysis.

- **Finding predictors** (Chap. 9)
  We have a special interest in the prediction of some target variable, but it (possibly) represents only one building block of our full problem, so we do not really care about the *how* and *why* but are just interested in the best-possible prediction. We may apply methods from, for instance, classification or regression.

Tools for Data Science As already mentioned, the key to success is often the proper combination of data preparation and modeling techniques. Data analysis software suites are of great help as they reduce data formatting efforts and ease method linking. There is a (growing) list of commercial and free software suites and tools which we do not attempt to summarize here. Many online resources provide comparative summaries.

Although the choice of the software suite has considerable impact on the project time (usability) and can help avoid errors (because some of them are easily spotted using powerful visualization capabilities), the suites cannot take over the full analysis process. They provide at best an initial starting point (by means of analysis templates or project wizards), but in most cases the key factor is the intelligent combination of tools and background knowledge (regarding the project domain and the utilized tools). The suites exhibit different strengths, some focus on supporting the human data analyst by sophisticated graphical user interfaces, graphical configuration and reporting, while others are better suited for batch processing and automatization.

1.4 How to Read This Book

In the next chapter we will take a glimpse at the data science process by looking over the shoulder of Stan and Laura as they analyze their data (while only one of them actually follows the data science process). The chapter is intended to give an impression of what will be discussed in much greater detail throughout the book. The subsequent chapters follow the data science process stages: we analyze the problem first in Chap. 3 (project understanding) and then investigate whether the available data suit our purposes in terms of size, representativeness, and quality in Chap. 4 (data understanding). If we are confident that the data are worth carrying out the analysis, we discuss the data preparation (Chap. 6) as the last step before we enter the modeling phase (Chaps. 7–9). As already mentioned, data preparation is already tailored to the methods we are going to use for modeling; therefore, we have to introduce the principles of modeling already in Chap. 5. Deployment and monitoring is briefly addressed in Chap. 10. Readers who, over the years, have lost some of their statistical knowledge can (partially) recover it in Appendix A. The statistics appendix is not just a glossary of terms to quickly look up details but also serves as a book within the book for a few preparative lessons on statistics before delving into the chapters about data science.

Most chapters contain a section that equips the reader with the necessary information for some first hands-on experience using KNIME Analytics Platform. We have settled on KNIME Analytics Platform because it supports the composition of complex workflows in a graphical user interface. Appendix B provides a brief introduction to KNIME Analytics Platform. The workflows discussed in this book—and many more—are available for download at the book's website. We will continuously update this material and also provide examples using the popular data science languages R and Python.

References

1. Berthold, M.R.: Principles of guided analytics. KNIME Blog (2018)
2. Berthold, M., Hand, D.: Intelligent Data Analysis. Springer, Berlin (2009)
3. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: Cross Industry Standard Process for Data Mining 1.0, Step-by-Step Data Mining Guide. CRISP-DM consortium (2000)
4. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI Press/MIT Press, Menlo Park/Cambridge (1996)
5. Feynman, R.P., Leighton, R.B., Sands, M.: The Feynman Lectures on Physics. Mechanics, Radiation, and Heat, vol. 1. Addison-Wesley, Reading (1963)
6. Hand, D.: Intelligent data analysis: issues and opportunities. In: Proc. 2nd Int. Symp. on Advances in Intelligent Data Analysis, pp. 1–14. Springer, Berlin (1997)
7. Kepler, J.: Astronomia Nova, aitiologetos seu physica coelestis, tradita commentariis de motibus stellae martis, ex observationibus Tychonis Brahe. (New Astronomy, Based upon Causes, or Celestial Physics, Treated by Means of Commentaries on the Motions of the Star Mars, from the Observations of Tycho Brahe) (1609); English edition: New Astronomy. Cambridge University Press, Cambridge (1992)
8. Kepler, J.: Harmonices Mundi (1619); English edition: The Harmony of the World. American Philosophical Society, Philadelphia (1997)
9. Kurgan, L.A., Musilek, P.: A survey of knowledge discovery and data mining process models. Knowl. Eng. Rev. **21**(1), 1–24 (2006)
10. Marban, O., Menasalvas, E., Fernandez-Baizan, C.: A cost model to estimate the effort of data mining process (DMCoMo). Inf. Syst. **33**, 133–150 (2008)
11. Tamagnini, P., Schmid, S., Dietz, C.: How to automate Machine Learning. KNIME Blog (2018)
12. Thornton, C., Hutter, F., Hoos, H., Leyton-Brown, K.: Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In: KDD '13 Proc. 19th ACM SIGKDD, vol. 19, pp. 847–855 (2013)

Practical Data Science: An Example

2

Before talking about the full-fledged data science process and diving into the details of individual methods, this chapter demonstrates some typical pitfalls one encounters when analyzing real-world data. We start our journey through the data science process by looking over the shoulders of two (pseudo) data scientists, Stan and Laura, working on some hypothetical data science problems in a sales environment. Being differently skilled, they show how things should and should not be done. Throughout the chapter, a number of typical problems that data analysts meet in real work situations are demonstrated as well. We will skip algorithmic and other details here and only briefly mention the intention behind applying some of the processes and methods. They will be discussed in depth in subsequent chapters.

2.1 The Setup

Disclaimer The data and the application scenario used in this chapter are fictional. However, the underlying problems are motivated by actual problems which are encountered in real-world data science scenarios. Explaining particular applicational setups would have been entirely out of the scope of this book since, in order to understand the actual issue, a bit of domain knowledge is often helpful if not required. Please keep this in mind when reading the following. The goal of this chapter is to show (and sometimes slightly exaggerate) pitfalls encountered in real-world data science setups and not the reality in a supermarket chain. We are painfully aware that people familiar with this domain will find some of the encountered problems strange, to say the least. Have fun.

The Data For the following examples, we will use an artificial set of data sources from a hypothetical supermarket chain. The data set consists of a few tables, which have already been extracted from an in-house database:[1]

[1] Often just getting the data is a problem of its own. Data science assumes that you have access to the data you need—an assumption which is, unfortunately, frequently not true.

M.R. Berthold et al., *Guide to Intelligent Data Science*,
https://doi.org/10.1007/978-3-030-45574-3_2

- **Customers**—Data about customers, stemming mostly from information collected when these customers signed up for frequent shopper cards;
- **Products**—A list of products with their categories and prices;
- **Purchases**—A list of products together with the date they were purchased and the customer card ID used during checkout.

The Analysts Stan and Laura are responsible for the analytics of the southern and northern parts, respectively, of a large supermarket chain. They were recently hired to help better understand customer groups and behavior and try to increase revenue in the local stores. As is unfortunately all too common, over the years the stores have already begun all sorts of data acquisition operations, but in recent years quite a lot of this data has been merged—however, still without a clear picture in mind. Many other stores had started to issue frequent shopping cards, so the directors of marketing of the southern and northern markets decided to launch a similar program. Lots of data have been recorded, and Stan and Laura now face the challenge to fit existing data to the questions posed. Together with their managers, they have sat down and defined three data science tasks to be addressed in the following year:

- Differentiating the different customer groups and their behavior to better understand their impact on the overall revenue,
- Identifying connections between products to allow for cross-selling campaigns, and
- Helping design a marketing campaign to attract core customers to increase their purchases.

Stan is a representative of the typical self-taught data science newbie with little experience on the job and some more applied knowledge about the different techniques, whereas Laura has some training in statistics, data processing, and data science process planning.

2.2 Data Understanding and Pattern Finding

The first analysis task is a standard data science setup: customer segmentation—find out which types of customers exist in your database and try to link them to the revenue they create. This can be used later to care for clientele who are responsible for the largest revenue source or foster groups of customers who are underrepresented. Grouping (or *clustering*) records in a database is the predominant method to find such customer segments: the data are partitioned into smaller subsets, each forming a more coherent group than the overall database contains. We will go into much more detail on this type of data science methods in Chap. 7. For now it suffices to know that some of the most prominent clustering methods return one typical example for each cluster. This essentially allows us to reduce a large data set to a small number of representative examples for the subgroups contained in the database.

Table 2.1 Stan's clustering result

Cluster-id	Age	Customer revenue
1	46.5	€ 1,922.07
2	39.4	€ 11,162.20
3	39.1	€ 7,279.59
4	46.3	€ 419.23
5	39.0	€ 4,459.30

The Naive Approach Stan quickly jumps onto the challenge, creates a dump of the database containing customer purchases and their birth date, and computes the age of the customers based on their birth date and the current day. He realizes that he is interested in customer clusters and therefore needs to somehow aggregate the individual purchases to their respective "owner." He uses an aggregating operator in his database to compute the total price of the shopping baskets for each customer. Stan then applies a well-known clustering algorithm which results in five prototypical examples, as shown in Table 2.1.

Stan is puzzled—he was expecting the clustering algorithm to return reasonably meaningful groups, but this result looks as if all shoppers are around 40–50 years old but spend vastly different amounts of money on products. He looks into some of the customers' data in some of these clusters but seems unable to find any interesting relations or any reason why some seem to buy substantially more than others. He changes some of the algorithm's settings, such as the number of clusters created, but the results are similarly uninteresting.

The Sound Approach Laura takes a different approach. Routinely she first tries to understand the available data and validates that some basic assumptions are in fact true. She uses a basis data summarization tool to report the different values for the string attributes. The distribution of first names seems to match the frequencies she would expect. Names such as "Michael" and "Maria" are most frequent, and "Rosemarie" and "Anneliese" appear a lot less often. The frequencies of the occupations also roughly match her expectations: the majority of the customers are employees, while the second and third groups are students and freelancers, respectively. She proceeds to checking the attributes holding numbers. In order to check the age of the customers, she also computes the customers' ages from their birth date and checks minimum and maximum. She spots a number of customers who obviously reported a wrong birthday, because they are unbelievably young. As a consequence, she decides to filter the data to only include people between the ages of 18 and 100. In order to explore the data more quickly, she reduces the overall customer data set to 5,000 records by random sampling and then plots a so-called histogram, which shows different ranges of the attribute *age* and how many customers fall into that range. Figure 2.1 shows the result of this analysis.

This view confirms Laura's assumptions—the majority of shoppers is middle aged, and the number of shoppers continuously declines toward higher age groups. She creates a second histogram to better inspect the subtle, but strange, cliff at around age 48 using finer setting for the bins. Figure 2.2 shows the result of this analysis.

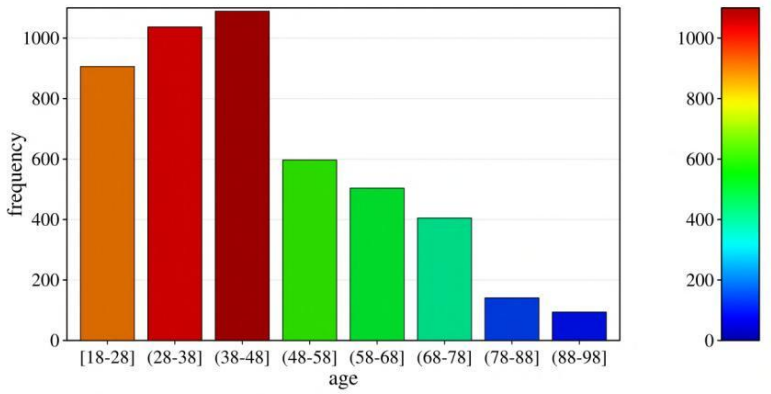

Fig. 2.1 A histogram for the distribution of the value of attribute *age* using 8 bins

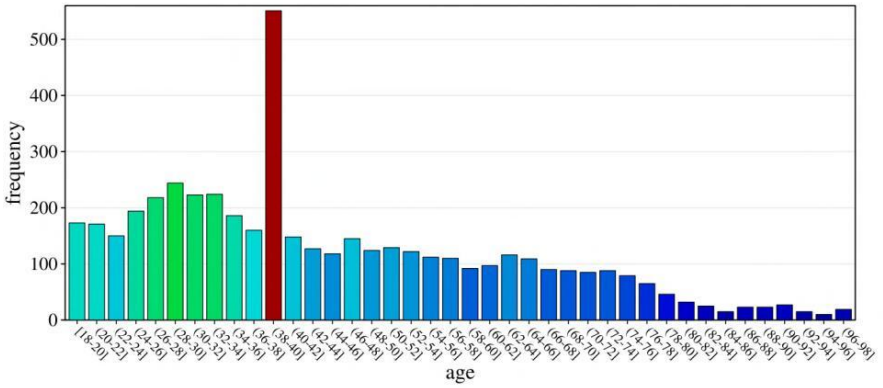

Fig. 2.2 A histogram for the distribution of the value of attribute *age* using 40 bins

Surprised, she notices the huge peak in the bin of ages 38–40. She discusses this observation with colleagues and the administrator of the shopping card database. They have no explanation for this odd concentration of 40-year-old people either (as of 2010). After a few other investigations, a colleague of the person who—before his retirement—designed the data entry forms suspects that this may have to do with the coding of missing birth dates. And, as it turns out, this is in fact the case: forms where people entered no or obviously nonsensical birth dates were entered into the form as zero values. For technical reasons, these zeros were then converted into the Java 0-date which turns out to be January 1, 1970. So these people all turn up with the same birth date in the customer database and in turn have the same age after the conversion Laura performed initially. Laura marks those entries in her database as "missing" in order to be able to distinguish them in future analyses.

Similarly, she inspects the shopping basket and product database and cleans up a number of other outliers and oddities. She then proceeds with the customer segmentation task. As in her previous data science projects, Laura first writes down her

Table 2.2 Laura's clustering result

Cluster	Age	Avg. cart price	Avg. purchases/month
1	75.3	€ 19.–	5.6
2	42.1	€ 78.–	7.8
3	38.1	€ 112.–	9.3
4	30.6	€ 16.–	4.8
5	44.7	€ 45.–	3.7

domain knowledge in form of a cognitive map, indicating relationships and dependencies between the attributes of her database. Having thus recalled the interactions between the variables of interest, she is well aware that the length of the customer's history and the number of overall shopping trips affect the overall basket price, and so she settles on the average basket price as a better estimator for the value of a particular customer. She also considers distinguishing the different product categories, realizing that those, of course, also potentially affect the average price. For the first step, she adds the average number of purchases per month, another indicator for the revenue a customer brings in. Data aggregation is now a bit more complex, but the modern data science tool she is using allows her to do the required joining and pivoting operations effortlessly. Laura knows that clustering algorithms are very sensitive to attributes with very different magnitudes, so she normalizes the three attributes to make sure that all three contribute equally to the clustering result. Running the same clustering algorithm that Stan was using, with the same setting for the number of clusters to be found, she gets the result shown in Table 2.2.

Obviously, there is a cluster (#1) of older customers who have a relatively small average basket price. There is also another group of customers (#4) which seems to correlate to younger shoppers, also purchasing smaller baskets. The middle-aged group varies wildly in price, however. Laura realizes that this matches her assumption about family status—people with families will likely buy more products and hence combine more products into more expensive baskets, which seems to explain the difference between clusters #2/#3 and cluster #5. The latter also seem to shop significantly less often. She goes back and validates some of these assumptions by looking at shopping frequency and average basket size as well, and also determines the overall impact on store revenues for these different groups. She finally discusses these results with her marketing and campaign specialists to develop strategies to foster the customer groups which bring in the largest chunk of revenue and develop the ones which seem to be underrepresented.

2.3 Explanation Finding

The second analysis goal is another standard shopping basket analysis problem: find product dependencies in order to better plan campaigns.

The Naive Approach Stan recently read in a book on practical data science how association rules can arbitrarily find such connections in market basket data. He

runs the association rule mining algorithm in his favorite data science tool with the default settings and inspects the results. Among the top-ranked generated rules, sorted by their confidence, Stan finds the following output:

```
'foie gras' (p1231) <- 'champagne Don Huberto' (p2149),
      'truffle oil de Rossini' (p578) [s=1E-5, c=75%]
'Tortellini De Cecco 500g' (p3456)'
   <- 'De Cecco Sugo Siciliana' (p8764) [s=1E-5, c=60%]
```

He quickly infers that this representation must mean that foie gras is bought whenever champagne and truffle oil are bought together and similarly for the other rule. Stan knows that the confidence measure c is important, as it indicates the strength of the dependency (the first rule holds in 3 out of 4 cases). He considers the second measure of frequency s to be less important and deliberately ignores its fairly small value. The two rules shown above are followed by a set of other, similarly luxury/culinary product-oriented rules. Stan concludes that luxury products are clearly the most important products on the shelf and recommends to his marketing manager to launch a campaign to advertise some of the products on the right-hand side of these rules (champagne, truffle oil) to increase the sales of the left side (foie gras). In parallel, he increases orders for these products, expecting a recognizable increase in sales. He proudly sends the results of his analysis to Laura.

The Sound Approach Laura is puzzled by those nonintuitive results. She reruns the analysis and notices the support values of the rules extracted by Stan—some of the rules Stan extracted have a remarkably high confidence, and some almost forecast shopping behavior. However, they have very low support values, meaning that only a small number of shopping baskets containing the products were ever observed. The rules that Stan found are not representative at all for his customer base. To confirm this, she runs a quick query on her database and sees that, indeed, there is essentially no influence on the overall revenue.

She notices that the problem of low support is caused by the fact that Stan ran the analysis on product IDs, so in effect he was forcing the rules to differentiate between brands of champagne and truffle oil. She reruns the analysis based on the product categories instead, ranks them by a mix of support and confidence, and finds a number of association rules with substantially higher support:

```
tomatoes <- capers, pasta  [s=0.007, c=32%]
tomatoes <- apples  [s=0.013, c=22%]
```

Laura focuses on rules with a much higher support measure s than before and also realizes that the confidence measure c is significantly higher than one would expect by chance. The first rule seems to be triggered by a recent fashion of Italian cooking, whereas the apple/tomato-rule is a known aspect.

However, she is still irritated by one of the rules discovered by Stan, which has a higher than suspected confidence despite a relatively low support. Are there some gourmets among the customers who prefer a very specific set of products? Rerunning this analysis on the shopping card owners yields almost the same results, so the (potential) gourmets appear among their regular customers. Just to be sure, she

inspects how many different customers (resp. shopping cards) occur for baskets that support this rule. As she had conjectured, there is a very limited number of customers that seem to have a strong affection for these products. Those few customers have bought this combination frequently, thus inflating the overall support measure (which refers to shopping baskets). This means that the support in terms of the *number of customers* is even smaller than the support in terms of *number of shopping baskets*. The response to any kind of special promotion would fall even shorter than expected from Stan's rule.

Apparently the time period in which the analyzed data have been collected influences the results. Thinking about it, she develops an idea how to learn about changes in the customers shopping behavior: She identifies a few rules, some rather promising, other well-known facts, and decides to monitor those combinations on a regular basis (say, quarterly). She got to know that a chain of liquor stores will soon open a number of shops close to the own markets, so she picks some rules with beverages in their conclusion part to see if the opening has any impact on the established shopping patterns of the own customers. As she fears a loss of potential sales, she plans a comparison of rules obtained not only over time but also among markets in the vicinity of such stores versus the other markets. She wonders whether promoting the products in the rule's antecedent may help bring back the customer and decides to discuss this with the marketing & sales team to determine if and where appropriate campaigns should be launched, once she has the results of her analysis.

2.4 Predicting the Future

The third and final analysis goal we consider in this brief overview is a forecasting or prediction problem. The idea is to find some relationship in our existing data that can help us predict if and how customers will react to coupon mailings and how this will affect our future revenue.

The Naive Approach Stan believes that no detailed analysis is required for this problem and notices that it is fairly straightforward to monitor success. At a competitor he has seen how discount coupons attract customers to purchase additional products. So he suggests launching a coupon campaign that gives customers a discount of 10% if they purchase products for more than €50. This coupon is mailed to all customers on record. Throughout the course of the next month, he carefully monitors his database and is positively surprised when he sees that his campaign is obviously working: the average price of shopping baskets is going up in comparison with previous months. However, at the end of the quarter he is shocked to see that overall revenues for the past quarter actually fell. His management is finally fed up with the lack of performance and fires Stan.

The Sound Approach Laura, who is promoted to head of analytics for the northern and southern supermarket chain, first cancels Stan's campaign and looks into the underlying data. She quickly realizes that even though quite a few customers did in fact use the coupons and increased their shopping baskets, their average number

of baskets per month actually went down—so quite a few people seem to have simply combined smaller shopping trips to be able to benefit from the discount offer. However, for some shoppers, the combined monthly shopping basket value did go up markedly, so there might be value here. Laura wonders how she can discriminate between those customers who simply use the coupons to discount their existing purchases and those who are actually enticed to purchase additional items. She notices that one of the earlier generated customer segments correlates better than others with the group of customers whose revenue went up—this fraction of customers is significantly higher than in the other groups. She considers using this very simple, manually designed predictor for a future campaign but wants to first make sure that she cannot do better with some smarter techniques. She decides that in the end it is not so important if she can actually understand the extracted model but only how well it performs.

To provide good starting points for the modeling technique, she decides to generate a few potentially informative attributes first. Models that rely on thousands of details typically perform poorly, so providing how often every product has been bought by the customer in the last month is not an option for her. To get robust models, she wants to aggregate the tiny bits of information, but what kind of aggregation could be helpful? She returns to her cognitive map to review the dependencies. One aspect is the availability of competitors: She reckons that customers may have alternative (possibly specialized) markets nearby but have been attracted by the coupon this time, keeping them away from the competitors. She decides to aggregate the money spent by the customer per month for a number of product types (such as *beverages*, thinking of the chain of liquor stores again). She conjectures that customers that perform well on average, but underperform in a specific segment only, may be enticed by the coupon to buy products for the underperforming segment also. Providing the segment performance before and after Stan's campaign should help a predictor detect such dependencies if they exist.

The cognitive map brings another idea to her mind: People who appreciate the full assortment but live somewhat further away from the own stores may see the coupon as a kind of travel compensation. So she adds a variable expressing a coarse estimation of the distance between the customer home and the nearest available market (which is only possible for the shopping card owners). She continues to use her cognitive map to address many different aspects and creates attributes that may help verify her hypotheses. She then investigates the generated attributes visually and also technically by means of feature selection methods.

After selecting the most promising attributes, she trains a classifier to distinguish the groups. She uses part of the data to simulate an independent test scenario and thereby evaluates the expected impact of a campaign—are the costs created by sending coupons to customers who do not purchase additional products offset by customers buying additional items? After some additional model fine-tuning, she reaches satisfactory performance. She discusses the results with the marketing&sales team and deploys the prediction system to control the coupon mailings for the next quarter. She keeps monitoring the performance of these coupon campaigns over future quarters and updates her model sporadically.

2.5 Concluding Remarks

In this chapter we have, very briefly and informally, touched upon a number of issues data scientists may encounter while making sense of real-world data. Many other problems can arise, and many more methods for data science exist in the academic literature and in real-world data science tools. We will attempt at covering the most prominent and most often used examples in the following chapters.

Note that one of the biggest problems data scientists very often have is that the data they get is not suited to answer the questions they are asked. For instance, if we were supposed to use the data in our customer database to find out how to differentiate Asian shopping behavior from European, we would have a very hard time. These data can only be used to distinguish between different types of European shoppers because it contains data from European markets only. Note also that we are (why ever) assuming that we used a nice, representative sample of all different types of European shoppers to generate the data—very often this is not the case, and the data themselves are already biased and will bias our analysis results—in this example we could be heavily biased by the type of supermarket chain we used to record the data in the first place. An upscale delicatessen supermarket will have dramatically different shopping patterns than a downscale discounter. We will be discussing these points later in more depth as well.

Project Understanding

3

We are at the beginning of a series of interdependent steps, where the project understanding phase marks the first. In this initial phase of the data analysis project, we have to map a problem onto one or many data analysis tasks. In a nutshell, we conjecture that the nature of the problem at hand can be adequately captured by some data sets (that still have to be identified or constructed), that appropriate modeling techniques can successfully be applied to learn the relationships in the data, and finally that the gained insights or models can be transferred back to the real case and applied successfully. This endeavor relies on a number of assumptions and is threatened by several risks, so the goal of the project understanding phase is to assess the main objective, the potential benefits, as well as the constraints, assumptions, and risks. While the number of data analysis projects is rapidly expanding, the failure rate is still high, so this phase should be carried out seriously to rate the chances of success realistically. The project understanding phase should be carried out with care to keep the project on the right track.

We have already sketched the data analysis process (CRISP-DM in Sect. 1.2). There is a clear order among the steps in the sense that for a later step, all precedent steps must have been executed. However, this does not mean that we can run once through all steps to deterministically achieve the desired results. There are many options and decisions to be made. Most of them will rely on our (subjective and dynamic) understanding of the problem at hand. The line of argument will not always be from an earlier phase to a later one. For instance, if a regression problem has to be solved, the analyst may decide that a certain method seems to be a promising choice for the modeling phase. From the characteristics of this technique he knows that all input data have to be transformed into numerical data, which has to be carried out beforehand (data preparation phase). This requires a careful look at the multivalued ordinally scaled attributes already in the data understanding phase to see how the order of the values is best preserved. If it is not considered in time, it may happen that later, in the evaluation phase, it turns out that the project owner expected to gain insights into the input–output relationship rather than having a black-box model only. If the analyst had considered this requirement beforehand, he might have chosen a different method. Changing this decision at any point later than in this initial

M.R. Berthold et al., *Guide to Intelligent Data Science*,
https://doi.org/10.1007/978-3-030-45574-3_3

Table 3.1 Problems faced in data analysis projects, excerpt from [1]

Problem source	Project owner perspective	Analyst perspective
Communication	Project owner does not understand the technical terms of the analyst	Analyst does not understand the terms of the domain of the project owner
Lack of understanding	Project owner was not sure what the analyst could do or achieve Models of analyst were different from what the project owner envisioned	Analyst found it hard to understand how to help the project owner
Organization	Requirements had to be adopted in later stages as problems with the data became evident	Project owner was an unpredictable group (not so concerned with the project)

project understanding phase often renders some (if not most) of the earlier work in data understanding, data preparation, and modeling useless. While the time spent on project and data understanding compared to data preparation and modeling is small (20% : 80%), the importance to success is just the opposite (80% : 20%) [4].

3.1 Determine the Project Objective

As a first step, a *primary* objective (not a long list but one or two) and some success criteria in terms of the project domain have to be determined (who will decide which results are desired and whether the original project goal was achieved or not). This is much easier said than done, especially if the analysis is not carried out by the domain expert himself. In such cases the project owner and the analyst *speak different languages* which may cause misunderstandings and confusion. In the worst case, the communication problems lead to very soft project goals, just vague enough to allow every stakeholder to see his own perspective somehow accounted for. At the end, all of a sudden the stakeholders recognize that the results do not fit their expectations. The challenge here is usually not a matter of technical but of communicative competence.

Table 3.1 shows some of the typical problems that occur in such projects. To overcome language confusion, a glossary of terms, definition, acronyms, and abbreviations is inevitable. Knowing the terms still does not imply an understanding of the project domain, objectives, constraints, and influencing factors. One interviewing technique that may help getting most out of the expert is to rephrase all of her statements, which often provokes additional relativizing statements. Another technique is to use explorative tools such as mind or cognitive maps to sketch beliefs, experiences, and known factors, and how they influence each other.

An example of a **cognitive map** in the shopping domain considered in Sect. 2 is given in Fig. 3.1. Each node of this graph represents a property of the considered product or the customer. The variable of interest is placed in the center: How often will a certain product be found in the shopping basket of the customer? This depends on various factors, which are placed around this node. The direction of influence is

in particular). Besides a plain listing of databases and personnel, it is important to clarify the access to both: If the data are stored in an operative system, mining the data may paralyze the applications using it. To become independent, it is advisable to provide a database dump. Experts are typically busy and difficult to grasp—but an inaccessible knowledge source is useless. A sufficiently large number of time slots for meetings should be arranged.

Based on the domain exploration (cognitive map, business process model, etc.), a list of explicit and implicit assumptions and risks is created to judge the chances of a successful project and guide the next steps. Data analysis lives on data. This list shall help convince ourselves that the data are meaningful and relevant to the project. Why should we undertake this effort? We will see whether we can build a model from these data later anyway. Unfortunately, this is only half of the truth. After reviewing a number of reports in a data analysis competition, Charles Elkan noted that "*when something surprising happens, rather than question the expectations, people typically believe that they should have done something slightly different*" [2]. Expecting that the problem can be solved with the given data may lead to continuously changing and "optimizing" the model—rather than taking the possibility into account that the data are not appropriate for this problem. In order to avoid this pitfall, the conjectured relations and expert-proven connections can help us verify that the given data satisfy our needs—or to put forward good reasons why the project will probably fail. This is particularly important as in many projects the available data have not been collected to serve the purpose that is intended now. To prevent us from carrying out an expensive project having almost no prospect of success, we have to carefully track all assumptions and verify them as soon as possible. Typical requirements and assumptions include:

- Requirements and constraints
  - *model requirements*,
    e.g., model has to be explanatory (because decisions must be justified clearly);
  - *ethical, political, legal issues*,
    e.g., variables such as gender, age, race must not be used;
  - *technical constraints*,
    e.g., applying the technical solution must not take more than n seconds;
- Assumptions
  - *representativeness*
    If conclusions about a specific target group are to be derived, a sufficiently large number of cases from this group must be contained in the database, and the sample in the database must be representative for the whole population.
  - *informativeness*
    To cover all aspects by the model, most of the influencing factors (identified in the cognitive map) should be represented by attributes in the database.
  - *good data quality*
    The relevant data must be of good quality (correct, complete, up-to-date) and unambiguous thanks to the available documentation.
  - *presence of external factors*

> We may assume that the external world does not change constantly—for instance, in a marketing project we may assume that the competitors do not change their current strategy or product portfolio at all.

Every assumption inherently represents a risk (there might be other risks though). If possible, a contingency should be sketched in case the assumption turns out to be invalid, including options such as the acquisition of additional data sources.

3.3 Determine Analysis Goals

Finally, the primary objective must be transformed into a more technical data mining goal. An architecture for the envisaged solution has to be found, composed of building blocks as discussed in Sect. 1.3 (data analysis tasks). For instance, this architecture might contain a component responsible for grouping the customers according to some readily available attributes first; another component finds interesting deviating subgroups in each of the groups; a third component predicts some variable of interest based on the customer data and the membership to the respective groups and subgroups. The better this architecture fits the actual situation, the better the chances of finding a model class that will prove successful in practice. To achieve this analogy, the discussions about the project domain are of great help.

Again there is the danger of accepting a reasonable architecture quickly, underestimating or even ignoring the great impact on the overall effort. Suppose that a company wants to increase the sales of some high-end product by direct mailing. One approach is to develop a model that predicts who will buy this product using the company's own customer database. Such a model might be interesting to interpret (useful for a report), but if it is used to decide to whom a mailing should be sent, most of the customers may have the product already (within the same customer database). Applying the model to people not being in the database is impossible as we lack the information about them that is needed by the model. The predictive model may also find out that customers buying the product were loyal customers for many years—but *artificially* increasing the duration of the customer relationship to support the purchase of the product is unfortunately impossible. If a foreseeable result is ignored or a misconception w.r.t. the desired use of the model is not recognized, considerable time may be wasted with building a correct model that turns out to be useless in the end.

For each of the building blocks, we can select a model class and technique to derive a model of this class automatically from data. There is nothing like *the unique best method for predictive tasks*, because they all have their individual weaknesses and strengths and it is impossible to combine all their properties or remove all biases (see Chap. 5). Although the final decision about the modeling technique will be made in the modeling phase, it should be clear already at this point of the analysis which properties the model should have and why. The methods and tools optimize the technical aspects of the model quality (such as accuracy, see also Chap. 5). Other aspects are often difficult to formalize and thus to optimize (such as interestingness

or interpretability), so that the choice of the model class has the greatest influence on these properties. Desirable properties may be, for instance:

- **Interpretability**
  If the goal of the analysis is a report that sketches possible explanations for a certain situation, the ultimate goal is to understand the delivered model. For some *black-box models*, it is hard to comprehend how the final decision is made, and their model lacks interpretability.
- **Reproduceability/stability**
  If the analysis is carried out more than once, we may achieve similar performance—but not necessarily similar models. This does no harm if the model is used as a black box, but hinders a direct comparison of subsequent models to investigate their differences.
- **Model flexibility/adequacy**
  A flexible model can adapt to more (complicated) situations than an inflexible model, which typically makes more assumptions about the real world and requires less parameters. If the problem domain is complex, the model learned from data must also be complex to be successful. However, with flexible models the risk of overfitting increases (will be discussed in Chap. 5).
- **Runtime**
  If restrictive runtime requirements are given (either for building or applying the model), this may exclude some computationally expensive approaches.
- **Interestingness** and **use of expert knowledge**
  The more an expert already knows, the more challenging it is to "surprise" him or her with new findings. Some techniques looking for associations (see Sect. 7.6) are known for their large number of findings, many of them redundant and thus uninteresting. So if there is a possibility of including any kind of previous knowledge, this may ease the search for the best model considerably, on the one hand, and may prevent us from rediscovering too many well-known artifacts.

When discussing the various modeling techniques in Chaps. 7–9, we will give hints which properties they possess. The final choice is then up to the analyst.

3.4 Further Reading

The books by Dorian Pyle [4, 5] offer many suggestions and constructive hints for carrying out the project understanding phase; [5] contains a step-by-step workflow for business understanding and data mining consisting of various *action boxes*. An organizationally grounded framework to formally implement the business understanding phase of data mining projects is presented in [6]. In [1] a template set for educing and documenting project requirements is proposed.

References

1. Britos, P., Dieste, O., García-Martínez, R.: Requirements elicitation in data mining for business intelligence projects. In: Advances in Information Systems Research, Education and Practice, pp. 139–150. IEEE Press, Piscataway (2008)
2. Elkan, C.: Magical thinking in data mining: lessons from coil challenge 2000. In: Proc. 7th Int. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 426–431. ACM Press, New York (2001)
3. Marban, O., Segovia, J., Menasalvas, E., Fernandez-Baizan, C.: Towards data mining engineering: a software engineering approach. Inf. Syst. **34**, 87–107 (2009)
4. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann, San Mateo (1999)
5. Pyle, D.: Business Modeling and Data Mining. Morgan Kaufmann, San Mateo (2003)
6. Sharma, S., Osei-Bryson, K.-M.: Framework for formal implementation of the business understanding phase of data mining projects. Expert Syst. Appl. **36**, 4114–4124 (2009)

Data Understanding

4

The main goal of data understanding is to gain general insights about the data that will potentially be helpful for the further steps in the data analysis process, but data understanding should not be driven exclusively by the goals and methods to be applied in later steps. Although these requirements should be kept in mind during data understanding, one should approach the data from a neutral point of view. Never trust any data as long as you have not carried out some simple plausibility checks. Methods for such plausibility checks will be discussed in this chapter. At the end of the data understanding phase, we know much better whether the assumptions we made during the project understanding phase concerning representativeness, informativeness, data quality, and the presence or absence of external factors are justified.

We first take a general look at single attributes in Sect. 4.1 and ask questions like: What kind of attributes do we have, and what do their domains look like? What is the precision of numerical values? Is the domain of an attribute stable over time, or does it change? We also need to assess the data quality. Methods and criteria for this purpose are introduced in Sect. 4.2.

Data understanding requires taking a closer look at the data. However, this does not mean that we must browse through seemingly endless columns of numbers and other values. In this way we would probably overlook most of the important facts. Looking at the data refers to visualization techniques (Sect. 4.3) that can be used to get a quick overview of basic characteristics of the data and enable us to check the plausibility of the data to a certain extent. Visualization techniques are suitable for the analysis of single attributes and of attributes in combination. Apart from the pure visualization, it is also recommended to compute simple statistical measures for correlation between attributes as described in Sect. 4.4.

Outliers, values, or records that are very different from all others should be identified with methods described in Sect. 4.5. They might cause difficulties for some of the methods applied in later steps, or they might be wrong values due to data quality problems. Missing values (see Sect. 4.6) can lead to similar problems as outliers, and by simply ignoring missing values we might obtain wrong data analysis results, so we must be aware of whether we have to deal with missing values and, if we have to, of what kind the missing values are.

M.R. Berthold et al., *Guide to Intelligent Data Science*,
https://doi.org/10.1007/978-3-030-45574-3_4

of children or the number of times a customer has ordered from an online shop in the last 12 months.

Sometimes, categorical attributes are coded by numerical values. For instance, the three possible values *food, drinks, nonfood* of the attribute *general product category* might be coded by the numbers 1, 2, and 3. However, this does not turn the attribute *general product category* into a (discrete) numerical attribute. We should bear this fact in mind for later steps of the analysis to avoid that the attribute is suddenly interpreted as a numerical attribute: it does not make sense to carry out numerical operations like computing the mean on such coded categorical attributes. For a discrete attribute, though, especially when it represents some counting process, it is meaningful to calculate the mean value, even though the mean value will usually not be an integer number. It is meaningful to say that on average the customers buy products 2.6 times per year in our online shop. But it does not make sense that the average *general product category* we sell is 2.6, which we might obtain when we simply compute the mean value of the products we have sold based on the numerical coding of the general product categories.

In contrast to discrete numerical attributes, a continuous attribute can—at least theoretically—assume any real value. However, such numerical values will always be measured and represented with limited precision. It should be taken into account how precise these values are. Drastic round-off errors or truncations can lead to problems in later steps of the analysis. Suppose, for instance, that a cluster analysis is to be carried out later on and that there is one numerical attribute, say X, that is truncated to only one digit right after the decimal point, while all other numerical attributes were measured and stored with a higher precision. When comparing different records, such truncation for the attribute X influences their perceived similarity and might be a dominating factor for the further analysis only for this reason. Truncation errors and measurements with limited precision should be distinguished from values corrupted with noise. The problem of noise will be tackled in the context of data quality in Sect. 4.2 and will also be discussed in more detail in Chap. 5.

Numerical attributes can have an *interval*, a *ratio*, or an *absolute scale*. For an interval scale, the definition of what zero means is more or less arbitrary. The date is a typical example for an attribute measured on an interval scale. There are calendars with different definitions of the time point zero. For instance, the Unix standard time, counted in milliseconds, has its time point zero in the year 1970 of the Gregorian calendar. The same applies to temperature scales like Fahrenheit and Celsius degrees, where zero refers to different temperatures. Certain operations like quotients are not meaningful for interval scales. For example, it does not make sense to say that a temperature of 21 °C is three times as warm as 7 °C.[1]

In contrast to this, a **ratio scale** has a canonical zero value and thus allows us to compute meaningful ratios. Examples of ratio scales are height, distance, or duration. Distance can be measured in different units like meters, kilometers, or miles. But no matter which unit we choose, a distance of zero will always have the same

[1] Such a statement may make sense, though, for the Kelvin temperature scale, because on this scale the temperature is directly proportional to the average kinetic energy of the particles—and it is meaningful to compute ratios of energies.

meaning. Especially ratios, which do not make sense for interval scales, are often useful for ratio scales: The quotient of distances is independent of the measurement unit, so the distance 20 km is always twice as long as the distance 10 km, even if we change the unit kilometers to meters or miles. Whereas for a ratio scale, only the value zero has a canonical meaning and the meaning of other values depends on the choice of the measurement unit, for an **absolute scale**, there is a unique measurement unit. A typical example for an absolute scale is any kind of counting procedure.

4.2 Data Quality

The saying "garbage in, garbage out" applies to data analysis just as it does to any other area. The results of an analysis cannot be better than the quality of the data, therefore we should be concerned about the data quality before we carry out any deeper analysis with the data. **Data quality** refers to how well the data fit their intended use. There are various data quality dimensions.

Accuracy is defined as the closeness between the value in the data and the true value. For numerical attributes, accuracy means how exact the value in the data set is compared to the true value. Noise or limited precision in measurements can lead to reduced accuracy for numerical attributes. Limited precision is often obvious from the data set. For example, in the Iris data set all numerical values are measured with only one digit right after the decimal point. The magnitude of noise can be estimated when measurements for the same value have been taken repeatedly. Accuracy of numerical values can also be affected by wrong or erroneous measurements or simply by errors like transposition of digits when measurements are recorded manually. For categorical attributes, problems with accuracy can result from misspellings like "fmale" for a value of the attribute *gender*, and also from erroneous entries.

We distinguish between *syntactic* and *semantic accuracy*. **Syntactic accuracy** means that a considered value might not be correct, but it belongs at least to the domain of the corresponding attribute. For a categorical attribute like *gender* for which only the values *female* and *male* are admitted, "fmale" violates syntactic accuracy. For numerical attributes, syntactic accuracy does not only mean that the value must be a number and not a string or text. Also certain numerical values can be out of the range of syntactic accuracy. Attributes like *weight* or *duration* will admit only positive values, and therefore negative values would violate syntactic accuracy. Other numerical attributes have an interval as their range like [0, 100] for the percentage of votes for a candidate. Negative values and values larger than 100 should not occur. For integer-valued attributes like the number of items a customer has bought, floating-point values should be excluded.

Problems with **semantic accuracy** mean that a value might be in the domain of the corresponding attribute, but it is not correct. When the attribute *gender* has the value *female* for the customer John Smith, then this is not a question of syntactic

accuracy, since *female* is a possible value of the attribute *gender*. But it is obviously a wrong value for a person named "John".[2]

Discovering problems of syntactic accuracy in a data set is a relatively easy task. Once we know the domains of the attributes, we can easily verify, whether the values lie in the corresponding domains or not. A simple measure for syntactic accuracy is the fraction of values that lies in the domains of their corresponding attribute.

The verification of semantic accuracy is much more difficult or often even impossible. Another source for the same data would enable us to check our data, and differences not caused by problems with syntactic accuracy indicate problems with semantic accuracy. Sometimes also certain "business rules" are known for the data. For instance, if we find a record in our data set with the value *male* for the attribute *gender* and *yes* for the attribute *pregnant*, there must be a problem of semantic accuracy based on the known "business rule" that only women can be pregnant.

Whether or to what detail we check syntactic and semantic accuracy depends very much on how the data were generated. Especially, when data were entered manually, there is a higher chance of accuracy problems. In any case, it is recommended to carry out at least some simple tests to see whether there might be problems with accuracy. However, the usual practice is to keep these tests at a minimum and to find out later on that there are problems with accuracy, namely when the data analysis yields implausible results.

Throughout this book we normally assume that the data are already given, for example, as a database table. This is not the best point in time to cope with data quality problems. Chances of avoiding or reducing data quality problems are highest when the data are entered into the database. For instance, instead of letting a user type in the value of categorical attribute with the danger of misspellings, one could provide a fixed selection of values from which the user can choose.

Another dimension of data quality is **completeness** which can be divided into completeness with respect to attribute values and completeness with respect to records. Completeness with respect to attribute values refers to missing values (which will be discussed in Sect. 4.6). When missing values are explicitly marked as such, then a simple measure for this dimension of data quality is the fraction of missing values among all values. But we will see that missing values are not always directly recognizable, so that the fraction of known missing values might only provide a lower bound for the fraction of actually missing values.

Completeness with respect to records means that the data set contains the necessary information that is required for the analysis. Some records might simply be missing for some technical reasons. Data might have been lost because a few years ago the underlying database system was changed and only those data records were transferred to the new database that were considered to be important at that point in time. In a customer database, customers who had not bought anything for a longer time might not have been transferred to the new database (in order to

[2]Note, however, that the problem may also reside with the name. Maybe the name of the person was misspelled, and the correct name is "Joan Smith"—then the gender is actually *female*.

eliminate potential zombie customers), or older transactions were not stored anymore.

Very often the available data set itself is biased and not representative. Consider as an example a bank that provides mortgages to private customers. If the aim of the analysis is to predict whether future applicants of loans will return the loan, we must take into account that the sample is biased in the sense that we only have information about those customers who have been granted a loan. For those customers who have been denied the loan initially, we have no information whether they would have returned the loan or not. But these customers especially could be the ones for whom it is interesting to find a good scheme to predict the risk. For customers with high income and a safe job and no current debt, we need no sophisticated data analysis techniques to predict that there is a good chance that they will return the loan. Of course, it is impossible to obtain a representative sample in the statistical sense in this case. Such a sample would mean that we would have to provide loans to any customer for a certain period, no matter how bad their financial status is, and collect and evaluate these data. Unfortunately, this would be a method entailing almost guaranteed bankruptcy.

The same problems occur in many other areas. For a production plant, we usually have large amounts of data when it is running in a normal mode. For exceptional situations, we will have little or no data. We cannot ask for such data, for instance, by requiring to check what happens if, say, a nuclear plant operates at its limit.

In such cases we might encounter future situations for which we had no corresponding data in our sample. Such possible gaps in the data should be identified. One should be aware that the space of possible values is automatically covered sparsely by the data when we have a larger number of attributes. Consider a set of m numerical attributes, and we want to make sure that we have at least positive and negative values for each attribute in our data set. This does not require a large data set. But if we want to make sure that we have data for all combinations of positive and negative values for the considered attributes, this leads to 2^m possible combinations. If we have $m = 20$ attributes, we already have more than one million possible combinations of positive and negative values. Therefore, if we have a data set with one million records, we have on average one sample for each of these combinations. For a data set with 100,000 records, at least 90% of the combinations will not be covered.

Other problems can be caused by **unbalanced data**. As an example, consider a production line for goods for which an automatic quality control is to be installed. Based on suitable measurements, a classifier is to be constructed that sorts out parts with flaws or faults. The scrap rate in production is usually very small, so that our data might contain far less than 1% examples for parts with flaws or faults.

Timeliness refers to whether the available data are too old to provide up to date information or cannot be considered as representative for predictions of future data. Timeliness is often a problem in dynamically changing domains, where only recently collected data provide relevant information, while older data can be misleading and can indicate trends that have vanished or even reversed.

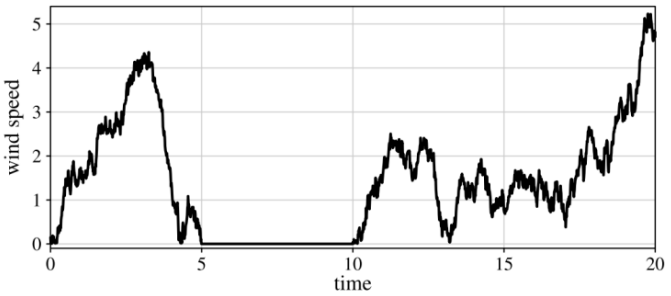

Fig. 4.1 Measured wind speeds with a period of a jammed sensor

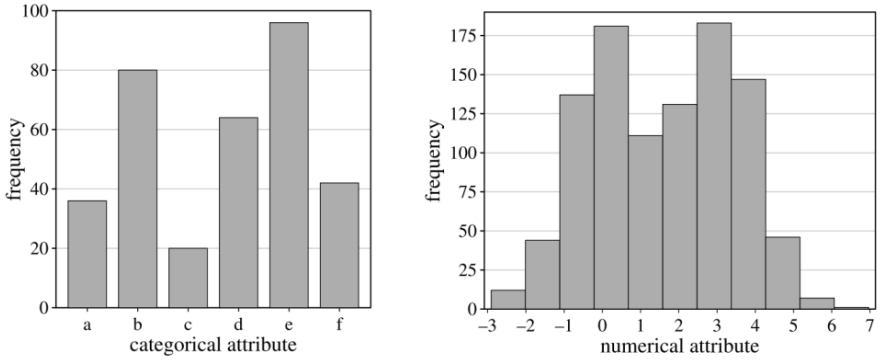

Fig. 4.2 A bar chart (categorical attribute, *left*) and a histogram (numerical attribute, *right*)

4.3 Data Visualization

According to Tukey [25], "there is no excuse for failing to plot and look" when one wants to handle a data analysis problem. The right plots of the data can provide valuable information as the simple time series plot in Fig. 4.1 shows, which enables us to discover zero values that are actually missing values. There are infinitely many ways to plot data, and it is not always easy to find the best ways of plotting a given data set. Nevertheless, there are some very useful standard data visualization techniques that will be discussed in the following.

4.3.1 Methods for One and Two Attributes

A **bar chart** is a simple way to depict the frequencies of the values of a categorical attribute. A simple example for a categorical attribute with six values a, b, c, d, e, and f is shown on the left in Fig. 4.2.

A **histogram** shows the frequency distribution for a numerical attribute. To this end, the range of the numerical attribute is discretized into a fixed number of intervals (called *bins*), usually of equal length. For each interval, the (absolute) frequency

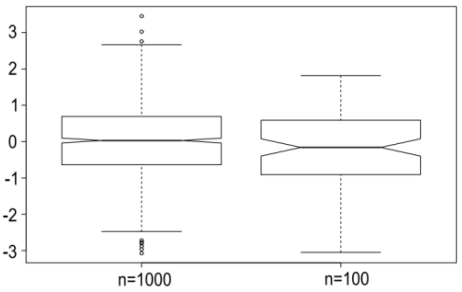

Fig. 4.6 Two boxplots for a sample from a standard normal distribution

extreme values from the sample (for instance, the 3% smallest and the 3% largest values) for calculating and displaying the histogram, or one can deviate from the principle of bins of equal length.

Boxplots are a very compact way to visualize and summarize main characteristics of a sample from a numerical attribute. Figure 4.6 shows two boxplots from samples from a standard normal distribution with mean 0 and variance 1. The left boxplot is based on sample of size $n = 1000$, whereas a sample of size $n = 100$ was used for the right boxplot.

The line in the middle of a boxplot indicates the sample median. The notch in the box is not always shown. It indicates a 95% confidence interval for the median. The box itself corresponds to the interquartile range covering the middle 50% of the data. The whiskers are drawn in the following way. The maximum length of each whisker is 1.5 times the length of the interquartile range. But if there is no data point at the maximum length of a whisker, the corresponding whisker is shortened until it reaches the next data point. Data points lying outside the whiskers are considered as outliers and are indicated in the form of small circles.

Comparing the two boxplots in Fig. 4.6, we can observe the following:

- Although both boxplots come from samples from the same normal distribution, they look different, since they are based on different samples.
- The notch of the left boxplot, representing a 95% confidence interval for the median, is much smaller than the notch of the right boxplot because of the larger sample size for the left boxplot.
- Theoretically, the whiskers for a sample from a symmetric distribution like the normal distribution should have roughly the same length. For the boxplot based on the smaller sample size, we can see that whiskers differ significantly in length, since—by chance—the largest value among the sample of 100 values was not greater than 2, whereas the smallest value was smaller than -3.
- In contrast to the boxplot on the left-hand side, the right boxplot does not contain any outliers. This is again due to the smaller sample size. The theoretical length of the interquartile range for a standard normal distribution is 1.349. Therefore, the probability of a point lying outside the (theoretical) range $[-2.698, 2.698]$ of the whiskers is almost 0.7%. Therefore, for a sample from a normal distribution of size $n = 1000$, we can expect roughly 7 outliers on average in a boxplot and less than one for a sample of size $n = 100$.

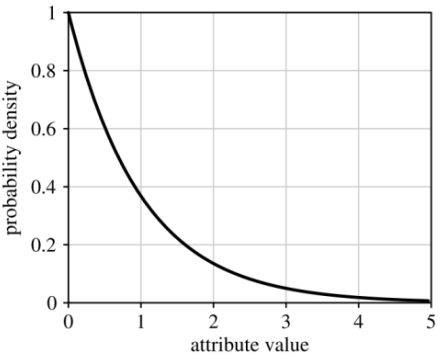

Fig. 4.7 The probability density function of the exponential distribution with $\lambda = 1$

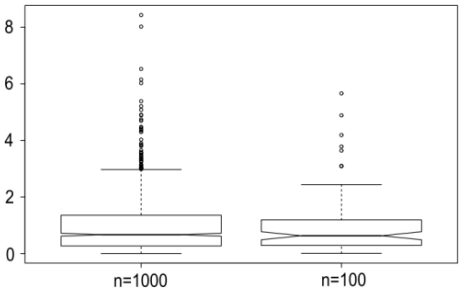

Fig. 4.8 Two boxplots for a sample from an exponential distribution

The boxplots of asymmetric distributions look completely different. If we sample from an exponential distribution, whose probability density function is shown in Fig. 4.7, we obtain boxplots as they are shown in Fig. 4.8. The boxplots on the left and right represent samples of sizes $n = 1000$ and $n = 100$, respectively.

Bar charts, histograms, and boxplots are visualizations for single attributes. In most cases, we have to deal with a number of attributes, and we are not only interested in the characteristics of single attributes but also in the relations and dependencies between the attributes. However, the display for visualizing the data is two-dimensional, and even if we use 3D-techniques from computer graphics, we cannot directly display more than two or three variables at the same time in a simple coordinate system, unless we use additional features such as symbols, color, and size. **Scatter plots** refer to displays where two attributes are plotted against each other. The two axes of the coordinate system represent the two considered attributes, and each instance in the data set is represented by a point, a circle, or any other symbol.

Simple scatter plots are not suited for larger data sets. For a data set with one million objects and a window size of 500×500 pixels, we would have on average four data objects per pixel. For larger data sets, many points or symbols in the scatter plot will be plotted at the same position, and we cannot distinguish whether a point in the scatter plot represents one or 100 objects. In the worst case, a scatter plot for a larger data set might simply look like the one in Fig. 4.9, providing only information about the range of the data, but no hint concerning the distribution of the data.

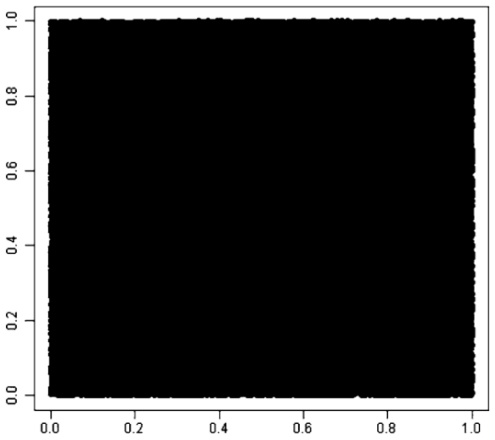

Fig. 4.9 A scatter plot of a data set with $n = 100{,}000$ instances

This can be amended by using **density plots** or plots based on **binning**. Using semitransparent points for plotting the data is one way to generate a density plot. Each plotted point is semitransparent, and the more points are plotted at the same place, the less transparent the image will become in this place. Binning was already used to generate histograms, and the principle is used for the scatter plots. The two-dimensional domain of the data for the scatter plot is partitioned into bins of the same size. Possible forms for the bins are rectangles or hexagons. The intensity of the color for the bin is chosen proportional to the number of data objects falling into the bin. Figure 4.10 shows a density plot on the left and a plot based on a hexagonal binning on the right for the same data set displayed in Fig. 4.9. Both plots indicate a higher density of the data around the point (0.6, 0.4), which cannot be seen in the simple scatter plot in Fig. 4.9.

Scatter plots can be enriched with further information in order to involve more attributes. Different plot symbols or colors can be used for plotting the points in order to include information about a categorical attribute. Color intensity and the size of the symbols are possible means to indicate the value of additional numerical attributes.

Figure 4.11 shows two scatter plots of the Iris data set—one displaying the sepal length versus the sepal width and the other one the petal length versus the petal width—in which different species are displayed by different colors. Both plots show that the red circles, representing the species Iris setosa, can be well distinguished from the other two species, Iris versicolor and Iris virginica, displayed as triangles and crosses, respectively. However, the left chart in Fig. 4.11 gives the impression that Iris virginica and Iris versicolor are very difficult to distinguish, at least when we only take the sepal length and the sepal width into account. When we consider the petal length and the petal width (right chart in Fig. 4.11), we can still see the overlap of the corresponding symbols for the species, but there is a clear tendency that Iris virginica tends to larger values than Iris versicolor for the petal length and width.

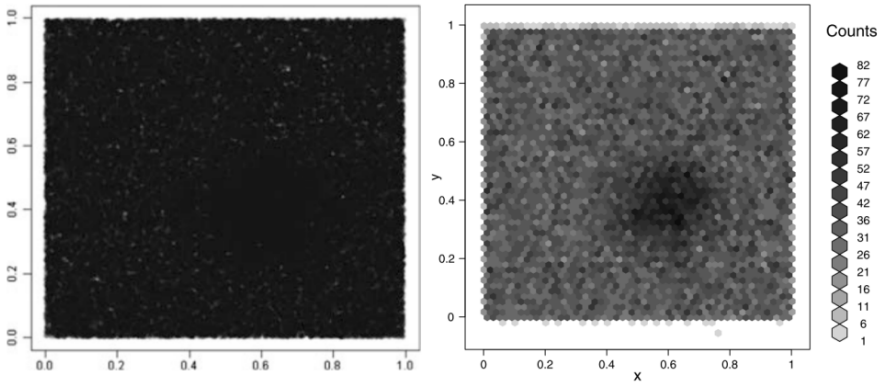

Fig. 4.10 Density plot (*left*) and a plot based on hexagonal binning (*right*) for the same data set as shown in Fig. 4.9

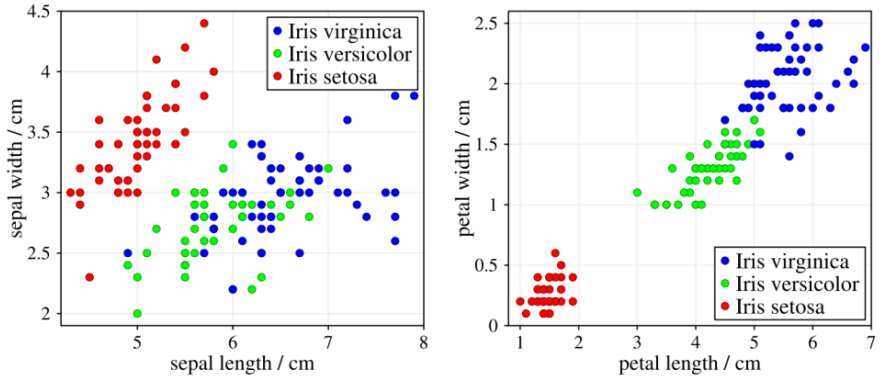

Fig. 4.11 Scatter plots of the iris data set for sepal length vs. sepal width (*left*) and for petal length vs. petal width (*right*). All quantities are measured in centimeters (cm)

Comparing the number of red circles in Figs. 4.11 (left and right), there seem to be less red circles on the right. But how can some of the objects suddenly vanish in the scatter plot? When we count the number of red circles, we see that in both scatter plots there are less than 50, although the data set contains 50 instances of Iris setosa that should be displayed by red circles. The circles are not missing in the scatter plots. Some circles are simply plotted at exactly the same position, since their measured sepal length and width or their measured petal length and width coincide. Recall that these values were only measured with a precision of just one digit right after the decimal point. To avoid this impression of seeing less objects than there actually are, one can add **jitter** to the scatter plot. Instead of plotting the symbols exactly at the coordinates specified by the values in the data set, we add a small random value to each original value in the data table. The left chart in Fig. 4.12 shows the resulting scatter plot with jitter where we have added random values from a uniform distribution on the interval $[-0.04, 0.04]$ to the original values. This ensures

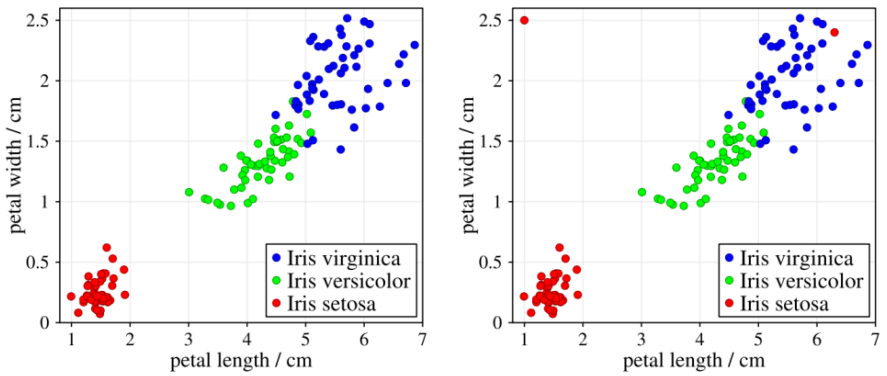

Fig. 4.12 The same scatter plot as in Fig. 4.11 on the right, but with jitter (*left*) and with jitter and two outliers (*right*; the outliers are the *red points* in the top left and top right corners)

that a point originally lying left or below another point will always remain left or below the other point, even when the jitter is added.

Jitter is essential when categorical attributes are used for the coordinate axes of a scatter plot, since categorical attributes have only a limited number of possible values, so that plotting of objects at exactly the same position occurs very often when no jitter is added.

From a scatter plot we can already extract important information. Consider again the scatter plot displayed in Fig. 4.12. We can see that the petal length and width are correlated. Objects with larger values for the petal length also tend to have larger values for the petal width. The scatter plot also shows that Iris setosa—the red circles in the scatter plot—can be easily distinguished from the other two species just on the basis of the petal length or width. The scatter plot does not indicate that the other two species cannot be separated clearly. It only shows that, solely based on the petal length and width, it is not possible to distinguish the two species perfectly. Outliers can also be discovered in scatter plots. The left chart in Fig. 4.12 does not have any outliers. In the right chart, however, we have added two artificial outliers. The data point in the upper left corner is a clear outlier with respect to the whole data set. Note that the values for the attributes petal length and width are both in the general range of the corresponding attributes in the data set. But there is no other object in the data set with a similar combination of these attribute values. The second outlier in the right chart of Fig. 4.12—the circle in the upper right corner—is not an outlier with respect to the values for the petal length and width or their combination. However, it is an outlier for the class Iris setosa displayed by red circles. Whenever such outliers are discovered, one should check the data or the data generating process again to ensure that the outliers are not due to erroneous data.

It should be noted that the scatter plots—like all other visualization techniques—are very useful tools to discover simple structures and patterns or peculiar deviations like outliers in a data set. But there is no guarantee that a scatter plot or any visualization technique will automatically show all or even any interesting or deviating

Table 4.1 Preservation of the variance of the Iris data set depending onthe number of principal components

	Principal component			
	PC1	PC2	PC3	PC4
Proportion of variance	0.73	0.229	0.0367	0.00518
Cumulative proportion	0.73	0.958	0.9948	1.00000

The left chart in Fig. 4.13 is a plot of the petal length and width of the Iris data set. However, apart from the necessary centering of the data around the origin by subtracting the mean, the data have been **z-score standardized** by the transformation

$$x \mapsto \frac{x - \hat{\mu}_X}{\hat{\sigma}_X}, \tag{4.6}$$

where $\hat{\mu}_X$ and $\hat{\sigma}_X$ are the mean value and empirical standard deviation of attribute X (see also Sect. 6.3.2), respectively. Without standardization, the result of PCA would depend on the scaling of the attributes. If no standardization is carried out, the attribute with the largest variance can easily dominate the first principal component. For the example in Fig. 4.13 with z-score standardization, the first principal component is the vector $(\sqrt{2}/2, \sqrt{2}/2)$. If the petal length is measured in meters instead of centimeters, but petal width is still measured in centimeters, the first principal component without z-score standardization becomes the vector $(0.0223, 0.9998)$, since the variance of the petal length has been decreased drastically by the scaling factor 0.01 resulting from the change from centimeters to meters, so that more or less only the petal width contributes to the variance in the data.

PCA can be used for visualization purposes by restricting to the first two principal components. More generally, PCA can carry out a dimension reduction to any lower-dimensional space; even more, PCA also provides information about over how many dimensions the data set actually spreads. This information can be extracted from the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_m$ of the covariance matrix. When we project the data to the first q principal components $v_1, \ldots, v_q$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_q$, this projection will preserve a fraction of

$$\frac{\lambda_1 + \cdots + \lambda_q}{\lambda_1 + \cdots + \lambda_m} \tag{4.7}$$

of the variance of the original data. Table 4.1 shows the corresponding result of PCA applied to the Iris data set without the categorical attribute for the species. A projection of this four-dimensional data set to the first principal component, i.e., to only one dimension, covers already 73% of the variance of the original data set. A projection to a plane defined by the first two principal components covers already 95.8% of the variance. This means that the four numerical attributes of the Iris data set are not located on a two-dimensional plane in the four-dimensional space but do not deviate too much from the plane defined by the first two principal components. The right chart in Fig. 4.13 shows the projection of the Iris data set to the first two principal components where PCA was carried out after the z-score standardization had been applied.