Felix Weber

Artificial Intelligence for Business Analytics

Algorithms, Platforms and Application Scenarios

Springer

Artificial Intelligence for Business Analytics

Felix Weber

Artificial Intelligence for Business Analytics

Algorithms, Platforms and Application Scenarios

Springer

Felix Weber
Chair of Business Informatics and Integrated Information Systems
University of Duisburg-Essen
Essen, Germany

ISBN 978-3-658-37598-0 ISBN 978-3-658-37599-7 (eBook)
https://doi.org/10.1007/978-3-658-37599-7

The translation was done with the help of artificial intelligence (machine translation by the service DeepL.com). A subsequent human revision was done primarily in terms of content.

This Springer Vieweg imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH, part of Springer Nature.
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Preface

With a purely academic background, it does surprise many newcomers what in practice is called business intelligence (BI) and business analytics (BA). The first naive thought of university graduates is certainly that really complex artificial intelligence (AI) and advanced machine learning (ML) models are applied in any larger companies. How else could one remain competitive if one does not already know before the consumer which color of the new car or cell phone model will be desired in the future and then play this out situation-dependently via all conceivable advertising channels?

I was very surprised to find that most of the "intelligence" done are just simple descriptive statistics.[1] One of Germany's largest retailers only collects sales reports on a weekly basis and does not even aggregate them from the ERP system they use but rather have the individual local branches manually compile the numbers and enter them into their in-house information system. Another retail company is just starting with the most basic analyses of the store's business based on simple metrics, such as the promotional share of sales, that is, the number of advertised products relative to non-advertised ones, per receipt. This sudden interest is probably due to the emerging competition from online retailers, such as Amazon, which only entered the German grocery market in 2017 with Amazon Fresh.[2]

Much of the analysis in the business environment is descriptive analysis. They compute descriptive statistics (i.e., counts, totals, averages, percentages, minimums, maximums, and simple arithmetic) that summarize certain groupings or filtered versions of the data, which are typically simple counts of some events. These analyses are mostly based on standard aggregate functions in databases that require nothing more than elementary school math. Even basic statistics (e.g., standard deviations, variance, p-value, etc.) are

[1] However, this observation holds true for many of the ideas, concepts, and recommended actions coming from the idealized world of research. The system architectures are more similar to monolithic mainframe architectures than to the state of the art of distributed service-oriented architectures – or the planning and execution of projects ignores the last decades of research in project management and instead uses, if at all, Microsoft Excel-based "planning."

[2] To be honest, it must also be said that the success of German retailers in the past would not have necessitated a deeper examination of these issues.

quite rare. The purpose of descriptive analytics is to simply summarize and tell you what happened: sales, number of customers, percentage of total sales with items that were advertised, page views, etc. There are literally thousands of these metrics – it is pointless to list them – but they are all just simple event counts. Other descriptive analytics can be results of simple arithmetic operations, such as share of voice, average response time, percentage index, and the average number of responses per post. All of this takes place in a majority of companies today and is mostly referred to as business intelligence. Most often, the term advanced analytics is used to describe the extension of this reporting to include some filters on the data before the descriptive statistics are calculated. For example, if you apply a geo-filter first for social media analytics, you can get metrics like average post per week from Germany and average post per week from the Netherlands. And you can display that data on a fancy map for all the countries you are active in. Then all of a sudden you can call it advanced analytics.

However, this rudimentary analytics is not enough for a competitive advantage over competitors. Especially if you suddenly have to compete with digital natives like Google, Amazon, or Alibaba. In the age of digitalization, however, this is a real challenge for many industries. Amazon has turned book retailing, and then retail itself, upside down. Google is suddenly entering the automotive market with self-driving cars, Uber is demoting industry giants in the automotive industry (Volvo and Toyota) to mere suppliers, and Airbnb is taking over a large market share in the hotel industry without its real estate. As different as these examples are, they are based not only on software and platforms but, more importantly, also on sophisticated analytics. Uber has a huge database of drivers, so as soon as you request a car, Uber's algorithm is ready to go – in 15 seconds or less, it matches you with the driver closest to you. In the background, Uber stores data about every ride – even when the driver has no passengers. All of this data is stored and used to predict supply and demand and set fares. Uber also studies how transportation is handled between cities and tries to adjust for bottlenecks and other common problems.

Essen, Germany Felix Weber

About the Aim of the Book

The aim of this book is not to train you as a data scientist or data analyst, nor will anyone be able to call themselves an expert in artificial intelligence or machine learning after reading it – even if some management consultants will do so. Instead, the book introduces the essential aspects of business analytics and the use of artificial intelligence methods in a condensed form. First of all, the basic terms and thought patterns of analytics from descriptive and predictive to prescriptive analytics are introduced in section “Categorisation of Analytical Methods and Models”. This is followed by the business analytics model for artificial intelligence (BAM.AI), a process model for the implementation of business analytics projects in section “Procedure Model: Business Analytics Model for Artificial intelligence (BAM.AI)”, and a technology framework, including the presentation of the most important frameworks, programming languages, and architectures, in Chap. 3. After an introduction to artificial intelligence in Chap. 2 and especially the subfield of machine learning, the most important problem categories are described, and the applicable algorithms are presented roughly but in an understandable way in section “Types of Problems in Artificial Intelligence and their Algorithms”. This is followed by a detailed overview of the common cloud platforms in section “Business Analytics and Machine Learning as a Service (Cloud Platforms)”, which enables quick implementation of a BA project. Here, the reader is provided with a guide that allows them to get an overview of the extensive offerings of the major providers. Finally, several application scenarios from different perspectives show the possible use of AI and BA in various industries as case studies section “Build or Buy?”.

Since the book definitely sees itself as an introduction and overview for decision-makers and implementers in IT and the related application domains, references to more in-depth literature are made in many places.

Essen, Germany Felix Weber

Contents

1 Business Analytics and Intelligence

Need for Increasing Analytical Decision Support

Globalization, a potentially emerging scarcity of resources, significantly increased complexity of markets and the rise of the BRICS countries are the biggest challenges for the leading industrialized countries in recent years. For these nations and the companies based there, the main task for the next decades is to utilize the existing production capacities much more efficiently and to ensure an environment for highly developed industrial products. To meet these challenges, the main focus is on subsidy policies and research activities on complex concepts, such as the "Digital Factory" [1], "Industry 4.0" [2], or, in general, "Intelligent Production Systems" [3]. In addition to this major change, another focus is on the introduction of a variety of systems to manage, optimize and control the existing operating processes. The main goal of these measures is to strive for the complete digitalization[1] and integration of all processes of the product life cycle, including the supply chains.

Analytics has become the technology driver of this decade. Companies like IBM, Oracle, Microsoft, and others have created entire organizational units focused solely on analytics to help the companies they advise work more effectively and efficiently. Decision-makers are using more computerized tools to help them do their jobs. Entire areas of operational management and administration could be replaced by automated analytical systems. And even consumers are using analytics tools directly or indirectly to make decisions about routine activities, such as shopping decisions (cue "price comparison tools"), healthcare (cue "health apps"), and entertainment (Netflix for example). Business analytics is rapidly evolving and increasingly focusing on innovative applications and use-cases for data that was not even captured until recently.

[1] In the sense of a digital representation of reality (also known as "Digital Twin").

F. Weber, *Artificial Intelligence for Business Analytics*,
https://doi.org/10.1007/978-3-658-37599-7_1

What companies need now, is a way for the right people to have the right data and information available at the right time, thus gaining a basis for rational decision-making that meets strategic and operational market conditions. And that is exactly the overall guideline and main requirement regarding analytics we propose in this book:

> Requirement: provide the right data to the right people at the right time for decision support.

The term decision support was deliberately chosen because data, information, or knowledge is provided to the user for a specific purpose: to facilitate decisions that have to be made one way or the other. Here, the striking example would be analyzing the cash register receipts in a grocery store and knowing when, which products, combinations and how often these are sold during a year. Knowing this enables those in charge, such as the store manager, a basis to support their decisions about shelf placement, replenishment, or price changes. These decisions have to be made one way or another, it is just that analytics makes it possible for these decisions not to be dependent on pure "gut feeling" or years of experience.

Regardless of whether predictive models are used, a company's historical data at least contains a clue as to why the company is in the current situation, as this data depicts past situations and decisions. Because of its technical proximity and complexity, analytics is most often viewed as a pure IT discipline, driven primarily by the company's technical environment. However, this categorization neglects the necessarity of domain-related knowledge. Thus, doing analytics without taking into account the organization (mission, vision, strategy, and goals) and the exact knowledge of the real business processes, which are mostly not documented or mapped in IT systems, will hardly lead to the optimal result. If we only look at the example of the store manager mentioned above, we can quickly derive a wealth of influencing factors that a pure IT focus would have neglected - also due to the fact that some data is not being recorded: demographics, socio-economic environment (local and macroeconomic), customer sentiments or even the peculiarities of local consumers (known to the store manager through his daily work but hardly ever mapped in an IT system).

A basic framework for analytics was and is actually present in every company, be it just the ubiquitous Excel files. In recent years, however, the underlying IT systems have undergone some important developments.

A big change is called "Big Data". In this context, the size[2] of the data is the first and sometimes the only dimension that stands out when big data is mentioned. At this point, we do not want to dive too deep into the origin and background of the term and only document the basic concept related to big data. Back in 2001, Laney [4] proposed that volume, variety, and velocity (or the three Vs) are the three dimensions of data management challenges. The 3Vs-Model has since evolved into a common framework for describing Big Data. Also, Gartner [5] defines the term in a similar way:

[2] What is actually meant is the mass or quantity.

> "Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enables enhanced insight, decision making, and process automation."

Volume refers to the size of the data. Large amounts of data are specified in several terabytes and petabytes. A terabyte stores as much data as would fit on 1500 CDs or 220 DVDs, enough to store about 16 million Facebook photos. Facebook processes up to one million photos per second [6]. A petabyte is equivalent to 1024 terabytes. However, the definitions of big data depend on the industry and the type of data and do not allow us to easily define a specific threshold for big data. For example, two data sets of the same size may require different technologies to be processed depending on the type (table vs. video data).

Variety refers to the structural heterogeneity in a data set. Modern technologies allow companies to use different types of structured, semi-structured, and unstructured data. Structured data refers to the tabular data in spreadsheets or relational databases. Text, images, audio, and video are examples of unstructured data that lack structural order but are required for some types of analysis. Across a continuum between fully structured and unstructured data, the format of semi-structured data does not meet strict standards on either side. Extensible markup language (XML) is a textual language for exchanging data on the web and is a typical example of semi-structured data. XML documents contain user-defined data tags that make them machine-readable.

Velocity refers to the speed at which data is generated and the speed at which it is to be analyzed and processed. The proliferation of digital devices, such as smartphones and sensors has led to an unprecedented rate of data creation and continues to drive an ever-increasing need for real-time analytics. Even conventional retailers generate high-frequency data; Wal-Mart, for example, processes more than one million transactions per hour [7].

Also, new technologies, such as in-memory databases (where the data is permanently located in the physical main memory of the computer) make it possible not only to process larger amounts of data but even in a shorter time. In a conventional database system, the data is disk-resident and the required data can be temporarily stored in the main memory for access and to be processed there, whereas in an in-memory database, the data is stored memory-resident and only as a backup copy on the hard disk - otherwise it remains fully in the main memory. In both cases, a given object can have copies both in memory and on disk. The main difference, however, is that with the in-memory database, the primary copy remains permanently in the main memory. As there has been a trend in recent years for main memory to become cheaper, it is now possible to move ever-larger databases into main memory. Since data can be accessed directly in memory, much better response times and transaction throughputs can be enabled. This is particularly important for real-time applications where transactions must be completed within specified time limits.

Also, the prevailing paradigm of software reference is currently changing. The increasing use of **cloud solutions** (where software and data are not hosted at the user site) tends to enable a shorter time-to-market and the possibility to create initial tests and prototypes with new technologies earlier.

The aforementioned changes in the availability of data, the greater volume of data, and the availability of new software and software reference models for storage and processing serve as the basis for another trend: the increased use of analytical models for the automated control of entire operational processes. Thus, the key step that leads us to speak of business analytics rather than business intelligence (see the elaboration in section "Distinction Between Business Intelligence and Business Analytics") is that of transferring decisions from people to systems. Here are some examples:

- In **purely digital processes,** such as omnichannel marketing, decisions have already been transferred to the IT systems today. Customer communication is sent directly from the system to the customers, based on the systemic assessment of the individual customer. Examples include Amazon's promotional emails or Netflix's recommendations. Based on the customer's data history, recommender systems optimize the communication with the customer. But also the trading of stocks and currencies is now almost completely automated and the algorithms of the different trading companies compete against each other. Of course, the most successful investor here is the one who uses the best algorithm.
- **Semiphysically digitized processes** are processes in which analytics is used, for example, to predict future demand and automatically reorder the necessary goods. In this case, too, the winner in the market will be the company that executes the processes using the best-optimized algorithms. The Internet of Things (IoT) is another new term that describes the mappability of previously purely physical processes through sensors and sensor data in all kinds of everyday objects. For example, there are dairy farmers who have their cows milked almost entirely automatically by robots. Humans are only called in when necessary, such as when cows are found to be ill and need treatment, which cannot be done by machines (‚yet?). For this purpose, a wide variety of sensors from the barn, the environment, and the individual animals themselves are used and evaluated.
- **Fully digitally controlled physical processes**, such as the use of robots in the automated production of goods or cars. These robots react to external physical input and algorithms decide on the necessary response. They must be able to make autonomous decisions based on algorithms, using speech and video recognition to understand the physiological environment in which they operate.

In recent years, a huge number of processes have been digitally mapped, digitized, or completely automated, and the manual decisions associated with them have disappeared. In many ways, we are seeing today what people expected during the "dot-com" era, which was all about the possibilities of new automated and digitized business processes that allowed companies to compete globally based on extremely scalable business models. Even then, new entrants like Amazon.com were redefining the sale of books by transforming a purely physical process (the bookstore around the corner) into a physically digitized process (buying physical goods online). Later, Apple and Amazon began producing physical devices to further increase the ability to consume content (books, music, and movies) over the internet and thus the degree to which digital value is created. Less noticed by the public, physical production processes have become increasingly digital. Over the last decade, more and more business processes have been digitized to the point where the nearest competitor is only an app away (Apple iBooks vs. Google Play Books). The market-leading app is often the one that is integrated on the leading platform, offers the best user experience, and contains recommendations optimized for the individual based on customer-related data analyses.

Since analytics is increasingly used in digital processes and these processes can also be automated by analytics as business analytics nowadays is also much more than decision support for humans within a company. It is also about providing data but above all about controlling digitized processes intelligently and automatically. Support for people is clearly moving into the background. Based on these findings the definition of business analytics is derived:

> Business analytics refers to (1) technologies, systems, methods, and applications that (2) capture, process, and analyze data to enable (3) decision support or automation that adds value to the business.

Whereby this definition is further elaborated on during the course of this book as:

1. The basis for all data acquisition, processing, analysis, and the decisions based on them are always IT systems: from the IoT sensors, which pass their data to the centralized or distributed systems for further processing, to the centralized ERP system, which implements the decisions. The methods are described in section "Types of Problems in Artificial Intelligence and Their Algorithms", followed by the technologies, systems, and applications.
2. Data collection, processing, and interaction are described in section "Business Analytics Technology Framework (BA.TF)" in a comprehensive technology framework.
3. Examples of decision support and automation can be found throughout the book, especially in Chaps. 3 and 4.

Distinction Between Business Intelligence and Business Analytics

The perspective taken - present to future - some of the previous remarks also apply to the more familiar term of business intelligence (BI). In the following, we want to explain once, in which the conceptualizations of the BI and the BA differ and where there an intersection can be found. Thus, there is a similar framework in the perspectives that can be taken in the context of analytics (see Table 1.1).

Table 1.1 Perspectives of analytics

	Past	Present	Future
Information	What happened? (reporting)	What is happening right now? (alerts)	What will happen? (projection)
Insights	How and why did something happen? (Modeling)	What is the best next step? (recommendations)	What will happen in the best and worst case? (prediction, optimization)

Different perspectives can be adopted within the framework of analytics. The leading structure is the temporal frame between the past and the future

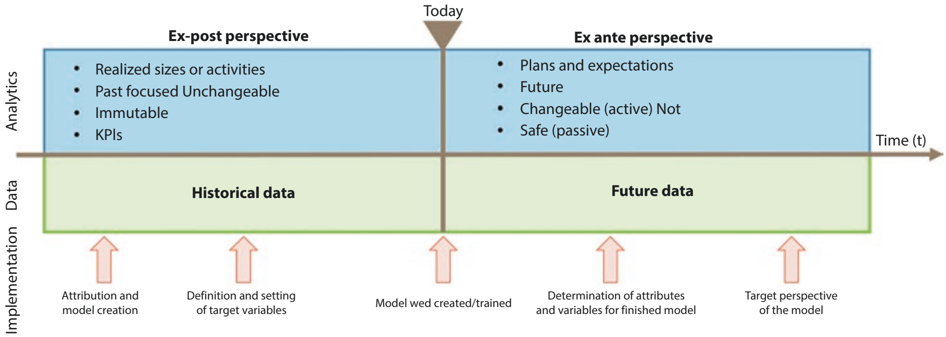

Today
Ex-post perspective
Ex ante perspective
Analytics
Data
Implementation
Realized sizes or activities
Past focused Unchangeable
Immutable
KPIs
Plans and expectations
Future
Changeable (active) Not
Safe (passive)
Time (t)
Historical data
Future data
Attribution and model creation
Definition and setting of target variables
Model wed created/trained
Determination of attributes and variables for finished model
Target perspective of the model

Fig. 1.1 The analytics, data, and implementation perspectives of business analytics (own illustration)

Most authors draw the line between business intelligence and business analytics in relation to the temporal objective of the different kinds of application. Thus, BI is generally assumed to have a purely ex-post and BA an ex-ante perspective. This demarcation is certainly correct from a technical point of view, for example, when looking purely at the algorithms used, as these are indeed based on key performance indicators (KPIs) formed by the aggregation of historical data. However, if we now extend the consideration to an overall perspective and look at BI from an overall corporate point of view, detached from the technical and operational level, BI certainly fits into a larger context of operational decision support. BI is never an end in itself but a support tool for decisions, carried out by people. Every KPI that is determined is basically only used to make an assessment and decision for future changes (or not) based on it. The KPI on sales by geographic sales region and the corresponding KPI on the change of the same can serve several purposes. Figure 1.1 illustrates the time-logic perspective with a timeline of data used to build predictive models or business intelligence reports. The vertical line in the middle represents the time when the model is built (today/now). The data used to build the models is on the left, as this always represents historical data - logically, data from the future cannot exist. When predictive models, which are the foundation of business analytics, are built to predict a "future" event, the data selected to build the predictive models will be based on a time before the date when the future event is expected. For example, if you want to build a model to predict whether a customer will respond to an email campaign, you start with the date the campaign was created (when all responses were received) to identify all participants. This is the date for the label "Definition and setting of the target variables" in Fig. 1.1. The attributes used as inputs must be known before the date of the mailing itself, so these values are collected to the left of the date the target variables were collected. In other words, the data is created with all the modeling data in the past but the target variable is still in the future until the date when the attributes are collected in the timeline of the data used for modeling. However, it is important to clarify that both business intelligence and business analytics analyses are based on the same data and the data is historical in both cases. The assumption is that future behavior to the right of the vertical line in Fig. 1.1. is consistent with the past behavior. If a predictive model identifies patterns in the past that predicts (in the past) that a customer will buy a product, the assumption is that this relationship will continue into the future – at least to a certain probability.

Automation as a Guiding Principle

As already indicated, BI and BA use fundamentally different methods of analysis (see section "Types of Problems in Artificial Intelligence and Their Algorithms") of the same data sets. However, with the different time perspective of the analysis results, BA enables a significantly different field of application, which we consider definitionally for the demarcation between BI and BA: with future models and forecasts, BA enables the shift of decision making from humans to IT systems. While in BI the results always have to be transferred from humans to the future (gut feeling or experiences are the basis for decisions here). This is not needed for any forecast generated by a BA system. Therefore, the subsequent processes can also be started automatically (e.g. start a new order from the wholesaler after predicting the expected future demand).

Categorisation of Analytical Methods and Models

Business analytics can range from simple reports to the most advanced optimization techniques (methods for determining the best possible course of action). Analytics can be divided into three broad categories: descriptive, predictive, and prescriptive analytics.

Descriptive Analytics

Descriptive analytics is known as the conventional approach to business intelligence and aims to present or "summarize" facts and figures in an understandable form to prepare data for communication processes or even further analysis by a human. Two primary techniques are used: data aggregation and "data mining"to identify past events. The goal here is to prepare historical data in an easily communicable format for the benefit of a broad business audience. A common example of descriptive analytics is company reports and KPIs that simply provide an overview of a company's operations, revenue, finances, customers, and stakeholders. Descriptive analytics helps describe and present data in a format that is easily understood by a variety of different audiences. Descriptive analytics rarely attempts to examine or establish cause and effect relationships. Because this form of analytics does not usually go beyond a cursory examination, the validity of results is easier to achieve. Some common methods used in descriptive analytics are observations, case studies, and surveys. Therefore, the collection and interpretation of large amounts of data in the big data environment can also play a role in this type of analytics, as it is relatively irrelevant how many individual values a KPI is aggregated from. Descriptive analytics is more suited to a historical presentation or a summary of past data and is usually reflected in the use of pure statistical calculations. Some common uses of descriptive analytics are:

- Creation of KPIs to describe the utilization of machines, delivery times, or waiting times of customers.
- Reports on market shares or related changes.
- Summary of past events from regional sales, customer churn, or marketing campaign success (click-through rates or cost/profit calculations).
- Tabular recording of social key figures, such as Facebook likes or followers.
- Reporting on general trends and developments (inflation, unemployment rate).

Predictive Analytics

Predictive analytics and statistics have a considerable overlap, with some statisticians arguing that predictive analytics is, at least at its core, just an extension of statistics [8]. For their part, predictive modelers often use algorithms and tests common to statistics. Most often, however, they do so without considering the (final) validation, which any statistician would do to ensure that the models are built “correctly” (validly). Nevertheless, there are significant differences between the typical approaches in the two fields. The table (Table 1.2) shows this clearly – maybe a bit to simplistic. Statistics is driven by theory, while predictive analytics does not follow this requirement. This is because many algorithms come from other fields (especially artificial intelligence and machine learning)

Table 1.2 Comparison between statistics and predictive analytics

Statistics	Predictive Analytics
Models are based on theory (there is an optimum)	Models are often based on non-parametric algorithms - no guaranteed optimum
Models are typically linear	Models are typically non-linear
Data sets are typically small; algorithms are often designed for accuracy with small data sets	Scaling particularly geared to large data sets; algorithms are not as efficient and stable for small data sets
The model is decisive	Data is crucial

Simplified comparison between statistics and predictive analytics and these usually do not have a provable optimal solution. But perhaps the most fundamental difference between the two fields (summarized in the last row of the table) is: for statistics, the model is the central element, while for predictive analytics it is the data.

Despite all the similarities between statistics and analytics, there is a difference in mindset that leads to differences in how analyses are performed. Statistics are often used to perform confirmatory analysis, where a hypothesis is made about a relationship between inputs and an output. And the purpose of the analysis is to confirm or deny the relationship and quantify the degree of that confirmation or denial.

Many analyses are highly structured, such as determining whether a drug reduces the incidence of a particular disease. Controls are essential to ensure that bias is not introduced into the model, which misleads the analyst's interpretation of the model. Coefficients of models are critical to understanding what the data is saying. Therefore, great care must be taken to transform model inputs and outputs to match the assumptions of the modeling algorithms. If a study predicts the effects of caloric intake, smoking, age, height, physical activity level, and metabolism on a person's weight, and one must trust the relative contribution of each factor on a person's weight, it is important to remove any bias due to the data itself so that the conclusions reflect the intent of the model. Bias in the data could lead the analyst to be misled that the inputs to the model have more or less influence than they do, simply because of numerical problems in the data. The residuals are also carefully examined to identify deviations from a normal distribution, although the requirement for normality decreases as the data size increases. If the residuals are not random with constant variance, the statistician will change the inputs and parameters until these problems are corrected.

Predictive modelers, on the other hand, often show little or only very superficial consideration for the final parameters in the models. The key is often the predictive accuracy of the model and thus its ability to inform and influence decisions. In contrast to the structured problem solved by confirmatory analytics using statistics, predictive analytics often attempts to solve less structured business problems using data that was not even collected for the purpose of model building but was only available by chance or as a byproduct.

Controls are often not present in the data and therefore causality, which is very difficult to uncover even in structured problems, will be extremely difficult to identify. For example, consider the process of analyzing which instance of a marketing campaign to run for an existing customer of a digital retailer. The customer might receive content from one of ten options identified by the email marketing group. Available modeling data includes the customer's master data, general (demographic), past website behavior and email interaction, and response to content played from one of the ten programs (clicks, dwell time, purchases made). The possible responses include ignoring, opening, and purchasing the advertised product or an alternative product. Predictive models can certainly be built to identify the best marketing approach of the ten possibilities on what to insert into the email based on a customer's behavior and demographics. However, this is anything but a controlled study. During the life of this marketing program, each customer continues to interact with their website, other websites, and see other promotions. The customer may have seen other display ads or performed Google searches in the meantime, further influencing their behavior. The purpose of such a model cannot be to fully understand why the customer behaves in a certain way because there are far too many unobserved influences, some of which are unobservable and some of which contradict each other. But this does not mean that the model is not useful.

Predictive modelers often use such unstructured problems and data, and the data, in whatever form it is available, is used in the models. This is not a problem as long as the data continues to be collected in a way that is consistent with the data as it was used in the models; the consistency of the data increases the likelihood that there will be consistency in the predictions of the model, and thus how well the model influences decisions. In many cases, this approach is simply due to the situation that controlled experiments, such as those most often conducted by statisticians, are often not feasible in the context of everyday business.

Prescriptive Analytics

The field of prescriptive analytics allows the user to “predict” a number of different possible actions and can guide them to an (optimized) solution. In short, these analytics are about advice. Prescriptive analytics attempts to quantify the impact of future decisions in order to advise on possible outcomes before the decisions are actually made. At its best, prescriptive analytics predicts not only what will happen but why it will happen and recommends actions that realize the benefits of the predictions.

These analyses go beyond descriptive and predictive analytics by recommending one or more possible courses of action. Essentially, they forecast multiple futures and allow companies to evaluate a range of possible outcomes based on their actions. Prescriptive analytics uses a combination of techniques and tools, such as algorithms, methods of artificial intelligence or machine learning, and modeling techniques. These techniques are applied to various data sets, including historical and transactional data, real-time data feeds, or big data.

Initiating, defining, implementing, and then using prescriptive analytics is relatively complex and most companies do not yet use them in their day-to-day business operations. When implemented properly, they can have a major impact on the way decisions are made, and thus on the company’s bottom line. Larger companies are successfully using prescriptive analytics to optimize production, planning, and inventory in the supply chain to ensure they deliver the right products at the right time and optimize the customer experience.

Business Analytics Technology Framework (BA.TF)

Frameworks play an important role in effective planning and allocation of resources. Also, frameworks can help an organization identify components and relationships between architectural components to understand an otherwise complex system structure. The frameworks for management information systems [9] and decision support systems [10] are early large frameworks that have helped organizations implement systems and make decisions. Scholars have also used them in mapping research trends and identifying research gaps.

Numerous frameworks have emerged in the course of information systems development, for example, the Zachman framework [11, 12] provides a common understanding regarding the integration of the components of a system regardless of its diversity, size, and complexity. In the field of decision support, the executive information systems (EIS) development framework [13] presents a structural perspective of EIS elements, their interaction, and the EIS development process. Since the work of the decision support systems (DSS) framework of [10], the variety of decision support frameworks has grown and matured [14] to include platforms for executive information systems, group decision support systems, geographic information systems, and more recently, business intelligence and big data.

The new framework introduced in this book (see Fig. 1.2) for business analytics with artificial intelligence methods, the "Business Analytics Technology Framework" (BA.TF), relies on existing frameworks of various related domains.

For example, Watson et al. (2007) divide the business intelligence framework [15] into two main activities: "getting data in" and "getting data out". The first part is traditionally referred to as data warehousing and involves the transfer and storage of data from a set of source systems into an integrated target system, the data warehouse. In the BA.TF, this is

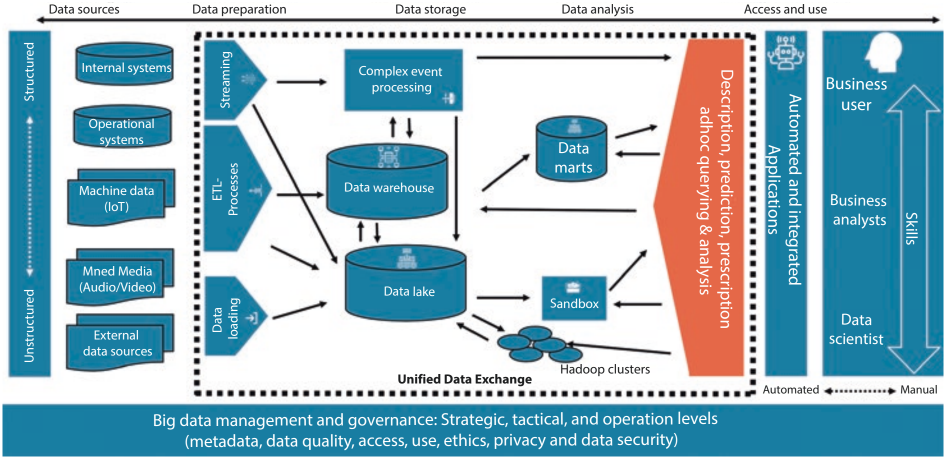


Fig. 1.2 Business Analytics Technology Framework (BA.TF) (own representation)

the left part, visualized with a right-rotated triangle. The sources can be within the company, provided by an external data provider, or from a business partner. Data sourcing is the most challenging aspect, requiring about 80 percent of the time and effort and accounting for more than 50 percent of the unexpected project costs [15]. The challenge here arises from poor data quality in the source systems (and that includes those internal to the organization and not necessarily external) and the use of legacy technology. In this context, data acquisition provides only limited value to an enterprise because the value can only be created from this data if users and applications can access it and use it to make decisions. This second activity, which we refer to here as analysis, model building, or automated decision making, is depicted in the BA.TF in the right-hand part, which is visualized with a left-turned triangle. Here, this part consists of data analysts, business users, and (automated) applications that access data to allow reporting, OLAP queries, and predictive and prescriptive analysis.

The Eckerson framework [16], for example, recognized "Business intelligence is changing" in 2011 and divides the necessary components of a BI architecture into the four underlying application purposes:

- Business intelligence needs to provide reports, dashboards, and scorecards, shown in BA.TF on the right side as manual access and usage.
- Analytical intelligence for "power users" and ad-hoc access to all data using spreadsheets, desktop databases, OLAP tools, data mining tools, and visual analysis tools.
- Continuous intelligence to collect, monitor, and analyze large volumes of rapidly changing data to support operational processes. Here, there should be near real-time delivery of information (i.e., hours to minutes) in a data warehouse to complex event processing and streaming systems that trigger alerts. These requirements can be found here in the area of complex event processing and streaming (see also section "Basic Concepts and Software Frameworks").
- Content intelligence enables the analysis of documents, web pages, email messages, social media pages, and other unstructured content. This has been addressed in the BA.TF, for example, through the use of the data lake.

The Industrial Internet Consortium (IIC)[3] has also defined a framework under the name "Analytics Framework" [17]. Although the focus here is very strongly oriented towards the industrial use of the Internet of Things, some generally applicable aspects can be found.

[3] The Industrial Internet Consortium (IIC) is an open membership organization with more than 250 members. The IIC says it was founded to accelerate the development, adoption and widespread use of interconnected machines and devices and smart analytics. Founded in March 2014 by AT&T, Cisco, General Electric, IBM and Intel, the IIC catalyzes and coordinates industrial internet priorities and enabling technologies with a focus on the Internet of Things.

Data Sources

Big data is also characterized by different types of data that can be processed for analysis. The left section "Data Sources" in the BA.TF shows which types of data are (can be) available to the organization.

Structured data still make up the majority of data used for analysis according to different surveys [18]. Structured data are mostly located in spreadsheets (Microsoft Excel), tables, and relational databases that conform to a data model that includes the properties and relationships between them. These data have known data lengths, data types, and data constraints. Therefore, they can be easily captured, organized, and queried because of the known structure. The BA.TF displays structured data from sources, such as internal systems that generate reports, operational systems that capture transactional data, and automated systems that capture machine data, such as customer activity logs.

Unstructured data comes in many different forms that do not adhere to traditional data models and are therefore typically not well suited for a relational database. Thanks to the development of alternative platforms for storing and managing such data, it is becoming increasingly common in IT systems. Unlike traditionally structured data, such as transactional data, unstructured data can be maintained in disparate formats. One of the most common types of unstructured data is plain text. Unstructured text is generated and stored in a variety of forms, including Word documents, email messages, PowerPoint presentations, survey responses, transcripts of call center interactions, and posts or comments from blogs and social media sites. Other types of unstructured data include images, audio, and video files. Machine data is another category that is growing rapidly in many organizations. For example, log files from websites, servers, networks, and applications - especially mobile - provide a wealth of activity and performance data. In addition, companies are increasingly collecting and analyzing data from sensors on production equipment and other devices connected to the Internet of Things (IoT).

In some cases, such data can be considered semi-structured - for example, when metadata tags are added to provide information and context about the content of the data. The boundary between unstructured and semi-structured data is not absolute. Semistructured data is even more widely used for analysis [18] because while this data does not have a strict and rigid structure, it does contain identifiable features. For example, photos and images can be tagged with time, date, creator, and keywords to help users search and organize them. Emails contain fixed tags, such as sender, date, time, and recipient attached to the content. Web pages have identifiable elements that allow companies to share information with their business partners.

Data Preparation

First of all, data preparation includes the classic processes of **extracting, transforming, and loading (ETL)** data and data cleansing. ETL processes require expert knowledge and are essential as a basis for analysis. Once data is identified as relevant, a human (for example, the team responsible for the data warehouse or data lake section "When to Use Which Algorithm?") extracts data from primary sources and transforms it to support the decision objective [15]. For example, a customer-centric decision may require that records from different sources, such as an operational transaction system and social media customer complaints, be merged and linked by a customer identifier, such as a zip code. Source systems may be incomplete, inaccurate, or difficult to access, so data must be cleansed to ensure data integrity. Data may need to be transformed to be useful for analysis, such as creating new fields to describe customer lifetime value (contribution margin a customer brings to the company throughout the relationship). The data can be loaded into a traditional data warehouse, a data lake, or more specific into Hadoop clusters (see section "When to Use Which Algorithm?"). Loading can be done in a variety of methods with a data warehouse either sequentially or parallel through tasks, such as overwriting existing data, updating data hourly or weekly.

For the use of a data lake, the ETL processes are not required, rather the data can be loaded directly into the application (**data loading**).

Lastly, depending on the deployment scenario, **streaming data** is also processed. This is data that is continuously generated by thousands of data sources that typically send in the data sets simultaneously and in small sizes (order of kilobytes). Streaming data includes a variety of data, such as log files created by customers using their mobile or web applications, e-commerce purchases, in-game player activity, information from social networks, financial trading floors, or geospatial services, and telemetry from connected devices or instruments in data centers.

This data must be processed sequentially and incrementally on a record-by-record basis or over rolling time windows and used for a variety of analyses, including correlations, aggregations, filtering, and sampling. The information derived from such analytics provides companies with insight into many aspects of their business and customer activities, such as service usage (for measurement and billing purposes), server activity, website clicks, and geolocation of devices, people, and physical assets, and enables them to respond quickly to new situations. For example, companies can track changes in public opinion about their brands and products by continuously analyzing social media streams and reacting in a timely manner when needed. Streaming data processing requires two layers: a storage layer and a processing layer. The storage layer must support record ordering and strong consistency to enable fast, low-cost, and replayable reading and writing of large data streams. The processing layer is responsible for consuming data from the storage layer, performing computations on that data, and then informing the storage layer to delete data that is no longer needed.

Data Storage

Traditionally, data is loaded into a data store that is subject-oriented (modeled by business concepts), integrated (standardized), time-varying (allows new versions), and non-volatile (unchanged and preserved over time) [19]. Therefore, data loading requires an established data dictionary[4] and data warehouse, which serves as a repository for verified data that the business uses for analysis. Data related to specific applications or departments can be grouped into a data mart[5] for ease of access or to restrict access. Moving and processing extremely large amounts of data as a monolithic dataset on a singular piece of hardware (server) is possible with today's technology up to a certain size but it is not very economical or practical. Therefore, storing and analyzing big data require dividing the processing among networked computers that can communicate with each other and coordinate actions in a distributed manner. Hadoop is an open-source framework that enables such distributed processing of data across small to large clusters. In this context, Hadoop is not an ETL tool but it supports ETL processes that run in parallel with and complement the data warehouse [20]. The results of the Hadoop cluster can be forwarded to the data warehouse or the analyses can be run directly on the clusters.

Depending on the requirements, a company needs both a data warehouse and a data lake, as the two concepts fulfill different requirements and use cases. For example, a data warehouse is first just a database that is optimized for analyzing relational data from transaction systems and business applications. The data structure and schema are defined in advance to optimize it for fast SQL queries, with the results typically used for operational reporting and analysis. Data is cleansed, enriched, and transformed so that it can act as a "single source of truth" that users can trust. A data lake differs from this because it stores relational data from the business application industry and non-relational data from mobile applications, IoT devices, and social media. The structure of the data or schema is not defined when the data is collected. This means that all data can be stored without any prior design or cleansing. The idea here is that data is simply stored as it arrives, and it is first secondary to what issues the data might be needed for in the future - but most of the time, with a data warehouse and ETL processes, this must necessarily be known in advance. Different types of analyses on the data are possible: SQL queries, big data analytics,

[4] A data dictionary contains metadata, that is, information about the database. The data dictionary is very important because it contains, for example, information about what is in the database, who is allowed to access it, where the database is physically located. The users of the database usually do not interact directly with the data dictionary, it is only managed by the database administrators, customized by the developers of the using applications and used in the context of the book by the analysts and data scientists.

[5] A data mart is a subset of data focused on a single functional area of an organization and stored in a data warehouse or other data store. A data mart is a set of all available data and has been tailored for use by a specific department, unit, or group of users in an organization (for example, marketing, sales, human resources, or finance).

full-text search, real-time analyses, and machine learning methods can all be used on a data lake. For a more in-depth technical description of the two concepts, see section "When to Use Which Algorithm?".

Analysis

The analysis encompasses a wide range of activities that can occur at different stages of data management and use [21]. Querying data is often the first step in an analytics process and is a predefined and often routine call to data storage for a specific piece of information; in contrast, ad hoc querying is unplanned and used when data is needed. Descriptive analytics is a class of tools and statistics used to describe data in summary form. For example, analysts may report on the number of occurrences of various metrics such as the number of clicks or number of people in certain age groups, or use summary statistics like means and standard deviations to characterize data. Descriptive analytics can use exploratory methods to try to understand data; for example, clustering can identify affinity groups. Exploratory analytics is often helpful in identifying a potential data element of interest for future studies or in selecting variables to include in an analysis. Predictive analytics refers to a group of methods that use historical data to predict the future of a particular target variable. Some of the most popular predictive methods are regression and neural networks. Prescriptive analytics is an emerging field that has received more attention with the advent of big data, as more future states and a wider variety of data types can be examined than in the past. This analysis attempts to explore different courses of action to find the optimal one by anticipating the outcome of different decision options [20].

Many of these processes have long been standard in data analytics. What differs with big data is the greater volume and variety of data being considered, and possibly the real-time nature of data collection and analysis. For example, Hadoop can be used to process and even store raw data from supplier websites, identify fraud-prone patterns and develop a predictive model in a flexible and interactive way. The predictive model could be developed on Hadoop and then copied into the data warehouse to find sales activity with the identified pattern. A fraudulent supplier would then be further investigated and possibly excluded [22]. As another example, graphical images of sale items could be analyzed to identify tags that a consumer is most likely to use to search for an item. The results can lead to improved labels to increase sales.

The "analytics sandbox"presented in the BA.TF is a scalable, development-oriented platform for data scientists to explore data, combine data from internal and external sources, develop advanced analytic models, and propose alternatives without changing the current data state of an enterprise. The sandbox can be a standalone platform in the Hadoop cluster or a logical partition in the enterprise data warehouse [23]. Traditional architectures use a schema-on-write and save-and-process paradigm, where data is first cleaned and prepared, stored, and only then queried. Complex event processing is proactive in-process monitoring of real-time events that enables organizations to make decisions and respond quickly to events, such as potential threats or opportunities [24]. The combination of real-time event processing, data warehousing, data lakes, data marts, Hadoop clusters, and sandboxing provides a data analytics and storage infrastructure that supports a stable environment while enabling innovation and real-time response.

> The analyses from section "Categorisation of Analytical Methods and Models", which can be carried out in detail, can be inserted into an organizational context (project) using the framework from section "Procedure Model: Business Analytics Model for Artificial intelligence (BAM.AI)" and implemented concretely using the algorithms presented in section "Reinforcement Learning" to solve the problems.

Access and Use

While building a BA system architecture happens only once, it becomes a daily business to use and query the data and conduct any form of analytics. In contrast to many other frameworks and in distinction to classical business intelligence (see section "Distinction Between Business Intelligence and Business Analytics"), the BA.TF distinguishes between two categories of data usage: automated applications and the integration of the analyses and models into the operational systems or, the manual usage of the data by users and analysts.

The Eckerson framework [16] categorizes manual use by two groups of users: casual users and power users. The casual users or occasional users are mostly executives, managers, and the core operational staff (accountants, sales people, customer service) and they use the basic functions of the systems. Reporting functionality can be used as needed or analytical processing can be integrated into the workflow of these users. For example, a call center representative talking to a customer can view the customer's KPIs, preferences, and potential offers for cross-selling. The power users are the analysts, analytics modelers, and data scientists who use the full power of the BI/BA systems available to them. They have a good knowledge of the system's functions, capabilities, and limitations, and a deep understanding of the business processes and the data behind those processes.

The framework distinguishes between three types of users: business users, business analysts, and data scientists. Business users have basic skills and domain-based knowledge. They include casual users in the context of Eckerson [16] but also external users, such as customers and suppliers who may connect through applications that depend on analytical processing. For example, an airline passenger creating and computing a multi-city itinerary may use a sophisticated scheduling application with a dynamic pricing engine without being aware of the complex processing involved. Business analysts are users who have more analytical skills than business users: they can analyze data and understand how data is organized. They use data retrieval via ad hoc queries, create specialized reports, build what-if scenarios, and interactively perform deeper analysis to support their decision-making. While these two roles roughly correspond to the two user types in the Eckerson framework [16], our framework identifies another type: data scientists as most advanced users. A data scientist has a strong background in mathematics, statistics, and/or computer science, an equally strong business acumen, and the ability to communicate with both business and IT leaders in ways that can influence how an organization addresses its business challenges using data. A data scientist can develop descriptive and predictive models (perhaps using the discovery platform; e.g. sandbox), evaluate models, and deploy and test them through controlled experiments. In the context of big data, data scientists may advise organizations on interpreting big data, managing big data, integrating data from multiple sources, and creating visualizations that facilitate understanding of data. They may also participate in communicating findings not only to the specialists and scientists on their team but also to business leaders and non-specialist audiences as needed.

(Big)-Data Management and Governance

Data management and governance should be an integral part of any organization and is imperative with the use of data in BI and BA initiatives [25]. Due to the increasing complexity of issues related to big data, organizations are facing new ethical, legal, and regulatory challenges in big data management and governance [26]. The data governance process must be balanced to meet top-down and bottom-up requirements [16]. The big data management and governance component identified in our framework propose a comprehensive data management approach that addresses problems at strategic, tactical, and operational levels. At the strategic level, a successful data governance process should encompass the entire spectrum from data acquisition to use and ensure that big data is in line with business strategy. Decisions include which internal and external data sources to use, selecting and deploying appropriate big data technologies for data storage and unified data sharing, and investing in training programs to have the appropriate skills to make informed and timely decisions. In the context of big data, organizations store more data than meets their immediate needs, which can expose them to more privacy and security risks. Adequate governance mechanisms to ensure compliance with regulations and laws are critical.

Unlike traditional BI, where most business units and users are provided with appropriate reports/data for decision-making in the context of big data, many organizations today are enabling their business units to find ways to leverage and analyze data to better meet their needs. Thus, it is not uncommon for big data projects to originate from different business units. Therefore, managing big data projects is critical. At a tactical level, the process of good governance should include ways to prioritize big data projects, establish metrics for evaluating projects and their benefits, and deploy knowledge management processes so that there is effective sharing of resources across the organization related to big data efforts.

Another big change in the context of big data and AI is the management overhead at the operational level. Latency (i.e., speed of data access) is critical. Since the data used by organizations is both internal and external, decisions need to be made at the operational level about how to handle data from different sources, such as how to structure unstructured data, how to ensure data quality (e.g., master data management), which in-memory databases to use for storage, and which no-SQL approaches to access data.

Procedure Model: Business Analytics Model for Artificial Intelligence (BAM.AI)

The most important thing in a large and complex project with a large number of participants and necessary competencies is to get an overview of the project from an overarching perspective as quickly as possible. Of course, this does not only apply to projects in the area of business analytics and the use of artificial intelligence methods presented in this book but it is especially true here. This is because it is imperative to have a project team with participants from at least the business department, IT, and additionally the implementing data scientists. In structuring these projects, this chapter focuses on the business analytics model for artificial intelligence (BAM.AI), which enables precisely such an overview and structuring.

The process model provides an overview of understanding and creating successful BA initiatives in any type of organization. The goal of the model is to provide the organization with a single common frame of reference for an overall structure in creating a successful BA project. To do this, the model clarifies the roles of each section and the interaction in the process of information generation and consumption. Setting up a BA project is a challenging task as the skills required to do so are very broad and require knowledge of the business domains, database technologies, data infrastructure, and predictive modeling algorithms and techniques. Very few people have all of these skills and therefore setting up a BA modeling project is necessarily a team effort. In doing so, this chapter describes a process model and principles for setting up a BA project.

If one follows one's own experience and that of numerous other authors, there is as yet no recognized standard for business analytics procedure models. Therefore, the work teams participating in such a project often have to develop their ad hoc methodology to manage the work dynamics within each team and between teams. The latter exhibits low process maturity and thus significantly increases the likelihood of failure.

Even though there are no uniform process models for BA projects, there are numerous methodologies for the subarea of data mining. Data mining was developed in the 1990s to discover data patterns in structured sources (databases) to gain added business value [27, 28] (see also the following definition). In this regard, there is a wide variation in the literature and practice in terms of problem domains, applications, formulations, and algorithms found in real-world applications. Therefore, "data mining" in itself is, first of all, a broad generic term describing these different aspects of data processing but in the vast majority of cases, it refers to the collection and analysis of data and less to the application of the gained knowledge in an automated perspective, as it is included in BA as a definitional basis (see also section "Distinction Between Business Intelligence and Business Analytics").

Data mining involves collecting, cleaning, processing, analyzing, and extracting useful insights from data sets [28].

In contrast to the original understanding of data mining, business analytics explicitly always takes unstructured data into account and works on a larger scale. A common point is that from a process perspective, both require close collaboration between data scientists and management to be successful.[6] Many methods and process models were originally developed for data mining and knowledge discovery. The first approach to this was knowledge discovery in a database (KDD) and this has been extended and refined into many other approaches. These approaches then became a kind of quasi-standard in the field of data mining under the name "Cross-Industry Standard Process for Data Mining" (CRISP-DM).

The business analytics model for artificial intelligence (BAM.AI) describes the BA process in clearly defined steps and represents a cross-industry process model, Fig. 1.3. It is closely based on the models that historically grew out of the KDD and is closely related to the CRISP-DM and the "Analytics Solutions Unified Method for Data Mining/Predictive Analytics"(ASUM-DM) developed by IBM [29]. One advantage of using BAM.AI is that it describes the most common steps in a generic way and enables the same understanding and approach within a company and in external communication.

The target audience of BAM.AI includes both managers and practitioners. For decision-makers and project managers, BAM.AI describes the steps in the modeling process from a program perspective and shows the steps that analysts need to take when setting up BA projects. For each of the steps, a separate cost and effort estimate can then also be

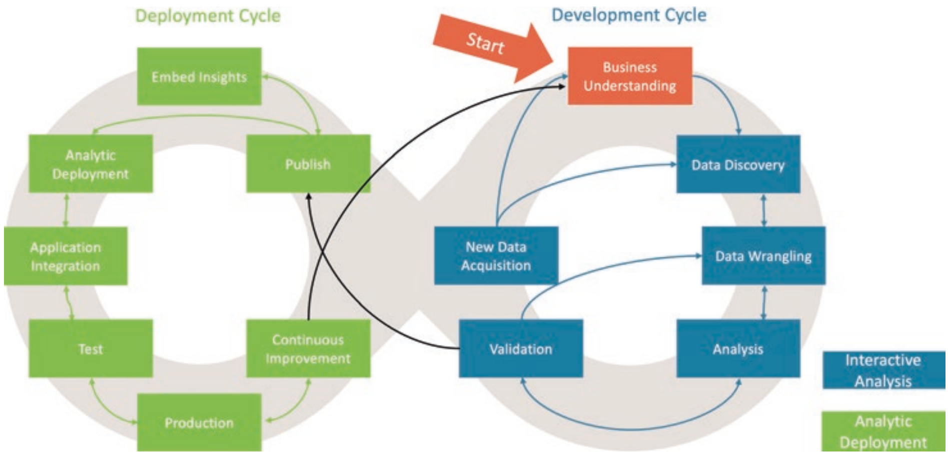


Fig. 1.3 Business Analytics (process) Model for Artificial Intelligence (BAM.AI)

[6] It should be noted that data mining now also takes unstructured data into account.

determined and tracked during the life of the project. In this way, a realistic estimate can be determined in advance and also kept transparent so that project deliverables and schedules are met. BAM.AI is divided into two cycles: the development cycle (section "Development Cycle") and the deployment cycle (section "Deployment Cycle").

For practitioners, both cycles together provide a structure for planning and implementing projects. Even for experienced practitioners, BAM.AI structures and describes the necessary steps concisely. There are good reasons for presenting the process model as two integrated circles, as many practitioners see the need to proceed in a non-linear manner (keyword "waterfall model") for such projects, as these projects are almost never run as planned due to problems with data (availability, structure, quality, and content) and the models (selection, training or performance). However, good structuring is always valuable, especially since BAM.AI provides a rationale for the steps that need to be taken in the business analytics process. BAM.AI combines practical experience with industry best practices to achieve successful, risk-minimized implementations. It draws on the collective experience of other process models in implementing software in the most demanding IT environments to meet a wide range of requirements. The two integrated cycles also correspond to the real conditions in most companies, where (project) development and actual operation are decoupled from each other. The process model, therefore, takes this reality into account but without forgetting that the data from productive operation forms the basis for improving the models. This is because AI methods are always dependent on large amounts of data for training (learning) - see Chap. 2 for more information.

Development Cycle

The first of the two cycles, the development cycle, focuses primarily on building the model and deriving it from historical data.

Business Understanding

Every BA project needs business objectives and domain experts who understand decisions, KPIs, estimates, or reports that are of value to a business and define the objectives of the project from a business perspective. The analysts used sometimes have this expertise themselves if they are hired in-house but domain experts usually have a better perspective on what aspects are important in the day-to-day business and how the project's results (should) impact the business. Without expertise, the goals set, definitions, what models should be built, and how they and the results from them should be evaluated can lead to failed projects that do not address the most important business issues. One way to understand the collaboration that leads to success in BA projects is to imagine a three-legged stool. Each leg is critical to ensuring that the chair remains stable and serves its purpose.

In BA projects, the three basic pillars are indispensable: (1) domain experts, (2) data or database experts, and (3) modeling experts (data scientists). The domain experts are required to formulate a problem comprehensively and in such a way that the problem is useful to the business. The data or database experts are needed to determine what data is available for modeling, how to collect more, qualitatively clean up the existing, and how to access that data. The modelers or data scientists are required to build the necessary models on this data and the defined questions.

If one or more of these three cornerstones are missing, then the problem cannot be defined properly or a purely technical view of the problem is taken (for example, if only modelers and the database administrator define the problems). Then it might be that excellent models are created with fantastic accuracy on the latest and hottest algorithms, but they cannot be used because the model does not meet the real needs of the business. Or in a more subtle way: maybe the model does support the right kind of decision but the models are scored in such a way that they do not address very well what is most important to the business - the wrong model is chosen because the wrong metric is used to evaluate the models.

On the other hand, if the database expert is not involved, then data problems may occur. First, there may not be enough understanding of the layout of tables in the databases to access all the fields needed for the algorithms. Second, there may be too little understanding of the individual fields and what information they represent, even if the names of the fields seem intuitive. Where honestly the normal state in many companies is that the names have been cryptically and arbitrarily created over years and no documentation is available. Third, insufficient permissions may prevent data from being used. Fourth, database resources may not support the type of joins or queries may exceed the available technical resources needed. And fifth, model deployment options envisioned by the BA team may not be supported by the organization. If data scientists are not available during this critical first phase, several obstacles may arise. First, a lack of understanding by project managers of what the algorithms and models can or should do. For example, managers, driven by the hype around AI, may specify specifications that are impossible to implement. Second, the definition of target variables for modeling may not be done at all or may be done poorly, hampering modeling efforts. Third, if the data scientist does not define the layout of the data needed to build the models, a data source to be used may not be defined at all, or crucial key fields may be missing but desperately needed for the models.

Data Discovery

In contrast to the CRISP-DM model, the new best practice approach defined here is to divide the data analysis process, also called data discovery, into two distinct steps:

Explore

(Data exploration) - After data has been prepared, this data is “explored” to see which parts of it help to find the answers we are looking for. In the process, the first tests can also be made and various hypotheses can be tested. You can also think of this step as data refinement or data selection. Companies and users can perform data exploration using a combination of automated and manual methods. Data scientists often use automated tools such as data visualization software for data exploration, as these tools allow users to quickly and easily point out the most relevant features and dependencies within a data set. In this step, users can identify the variables that seem most likely to yield interesting observations.

Discovery

(Data discovery) - once it is known what data is needed, it is possible to “dig deep” into that data to identify the specific points and variables that provide answers to the original question. It is a business-user-oriented process for identifying patterns and outliers through, for example, visual navigation through data or the application of automated advanced analytics. Discovery is an iterative process that does not require extensive upfront modeling. Data discovery requires skills around understanding data relationships and data modeling, in using data analytics and guided advanced analytics.

Data Wrangling

Data wrangling is one of those technical terms that are more or less self-explanatory: data wrangling is the process of cleansing, structuring, and enriching raw data into the desired format for better decision-making in less time. Data wrangling is increasingly ubiquitous among IT organizations and is a component of all IT initiatives. Data has become more diverse and unstructured, requiring more time to ingest, cleanse, and structure data for actual analysis. At the same time, business users have less time to wait for technical resources as data is required in almost every business decision - especially for analytics-focused organizations.

The basic idea of data wrangling is that the people who know the data and the real-world problems behind it best are the ones who investigate and prepare it. This means that business analysts, industry users, and managers (among others) are the intended users of data wrangling tools. In comparison, extract-transform-load (ETL) technologies focus on IT as the end-user. IT staff receive requests from their business partners and implement pipelines or workflows using ETL tools to deliver the desired data to the systems in the desired formats. Pure business users rarely see or use ETL technologies when working with data because they are not intuitive and are more at the database technology level than business operations. Before data wrangling tools were available, these users’ interaction with data was only in spreadsheets or business intelligence tools.

The process of data wrangling involves a sequence of the following processes:

- **Pre-processing,** which takes place immediately after data collection.
- **Standardization** of data into an understandable and usable format.
- **Cleaning** data from noise, missing or erroneous elements.
- **Consolidation** of data from different sources or data sets into a unified whole.
- **Comparison** of the data with the existing data records.
- **Filtering** of data through defined settings for subsequent processing.

Analysis

In the analysis phase, it is important to define what the solution must achieve in relation to the problem and environment. This includes both the features and the non-functional attributes (such as performance, usability, etc.). In the analysis phase, the modeling technique is first selected, then the necessary quality and validity of the model are determined, and finally, the model is implemented.

Modeling technique - In the first step of the analysis, the actual modeling technique, algorithms, and approaches to be used must be selected. Although a toolset or tool may have been selected previously, in the business understanding phase, the specific implementation must be determined in this phase. This is, of course, done with the original problem in mind. After all, whether you use decision trees or neural networks has a lot to do with the question to be answered, the available data, and the other framework conditions (which are set by the two phases beforehand). A detailed list of problem types and the algorithms that can be used in each case follows in section "Reinforcement Learning". It is always important to take into account, even if several techniques have to be used:

- **Modeling technique** - selection and definition of the modeling technique to be used.
- **Modeling assumptions** - Many modeling techniques make specific assumptions about the data, e.g., that all attributes are subject to equal distribution, no missing values are allowed, or class attributes must be symbolic.

Quality assurance - Before the actual model is created afterward, an additional procedure or mechanism must be defined to test the quality and validity of the model. For example, in supervised algorithms such as classification, it is common to use error rates as quality measures for the models. Therefore, when applying these algorithms, one typically divides the dataset into training and testing datasets (also called learning and validation datasets). In this process, the model is built on the training data and then the quality is estimated on the separate test set. It is also important at this stage to define the intended plan for training, testing, and evaluating the models. An essential part of the plan is to determine how the available data set will be divided into training, testing, and validation data sets.

Create models - The modeling tool (section "When to Use Which Algorithm?") is then applied to the prepared data set to create one or more models. This involves defining the necessary parameter settings. With most modeling tools, there are a large number of parameters that can be adjusted. In this process, the parameters and the selected values should be documented along with a rationale for the choice of parameter settings. Thus, the result of this substep includes both the model and the model description and documentation:

- **Models** - These are the actual models created with the modeling tool.
- **Model descriptions** - description of the resulting models, report on the interpretation of the models, and documentation of any difficulties and assumptions made in the creation process.

Validation

The models that have now been created must be interpreted according to the domain knowledge (preferably with the involvement of the business users in workshops), based on the previously defined success criteria and the test design. First, the results of the models created must be assessed technically (in terms of the quality criteria of the algorithms), and then the results must be discussed with the business analysts and domain experts in the business context. However, the assessment should not only measure the technical quality criteria but also take into account the business goals and business success criteria as much as possible. In most projects, a single technique is applied more than once and results are produced using several different techniques or in several steps. The procedure of technical evaluation (in the sense of mathematical quality assessment) can be divided into two steps:

- **Model evaluation** - A summary of the results of this step is provided, listing the goodness and quality of each generated model (e.g. in terms of accuracy). Then, the models are selected with respect to the quality criteria.
- **Revised parameter settings** - Depending on the model evaluation, the parameter settings must be adjusted again and tuned for the next modeling run. In doing so, model creation and evaluation should be iterated until the best model is found. It is important to document all changes, revisions, and evaluations.

The previous step of this phase addressed factors, such as the accuracy and generality of the model. The second step assesses the extent to which the model achieves the business objectives and whether there is a business reason why this model is sufficient or perhaps inadequate. Another option is to test the model(s) on trial applications in real-world use - if time and budget constraints permit. The evaluation phase also includes assessing any other deliverables that have been generated.

New Data Acquisition

Depending on the results, another iteration is now initiated, new necessary or possible data is determined and the model is looked at again. An inventory of the remaining resources and the budget should also be made if a fixed budget framework exists, as these restrictions can influence further decisions.

Deployment Cycle

The second of the two cycles, the deployment cycle, focuses primarily on the use and productive exploitation of the previously created model and its application to actual data. Most data scientists have little or no awareness of this other half of the problem. Many companies are struggling with BA and AI and the various pitfalls as shown by various studies [30, 31]. According to analysts, it takes about 2 months to take a single predictive model from creation to production. But why is it so difficult to scale BA and AI projects in an organization?

The productive implementation and maintenance of these projects is no easy task and most data scientists do not see it as their job. Yet the crucial questions are essential for the success of the project:

- How do I integrate the model or project into the existing system landscape?
- How can the model be easily deployed so that it can be consumed by other applications in a scalable and secure manner?
- How to monitor the quality of the models and release a new version if needed?
- How can the handover of artifacts from the data scientist to IT operations be managed without friction? Is this separation even necessary?

Publish

To determine where to productively run the analytics model created, the following considerations should be taken into account:

Scope - Ultimately, the scope is the derived information and decisions (not the raw data) and how to act on it that determines what types of analytics are used and where. For example, if the goal is to optimize machine availability at one location, analysis of the data collected there may be sufficient. In this case, the analysis can be performed anywhere, provided that normal local operations are not critically dependent on network latency and the availability of the analysis results. On the other hand, if the value proposition is to optimize production across sites that require comparison of factory efficiency, then analysis of data collected from those sites must be performed to be available at a higher level of the system architecture.

Response time and reliability - In an industrial environment, some problems require real-time analytics, calculations, and decisions, and others can be performed after the fact. The former almost always require analytics to be local for reliability and performance.

Bandwidth - The total amount of data generated (from sensors, for example), along with the data captured by the control or transaction systems, can be enormous in many cases. This must be factored into the overall increase in network and infrastructure utilization, depending on the location of the deployment.

Capacity - In some cases, it may be optimal to perform analytics at a particular level in a system architecture but the existing infrastructure may not be able to support it, so another level is selected.

Security - The value from data transfer must be balanced with concerns about transferring raw data outside of controlled areas and the associated costs. It may be more efficient to perform some analysis locally and share necessary summaries, redacted, or anonymized information with other areas. In the vast majority of cases, this discussion leads to a decision between a local or cloud-based location of deployment. The important thing here is to do an honest assessment (Are your on-premises admins really as good as the security experts at Amazon AWS or Google?).

Compliance - To illustrate how compliance can impact analytics as a design consideration, national security is used as an example. National security concerns can limit architectural decisions about data management and sharing with government regulations in industries, such as aerospace and defense. This will impact where analytics must be deployed to meet regulatory requirements, such as preventing large-scale computations from being performed in a public cloud facility to reduce costs.

Platform - When it comes to deployment, a Platform as a Service (PaaS) or Infrastructure as a Service (IaaS) must be chosen. A PaaS may be suitable for prototyping and businesses with lower requirements. As the business grows and/or higher demands are made, IaaS is probably the better way to go. This requires handling more complexity but allows scaling to be implemented much better (and probably cheaper). There are many solutions available from the big hyper scalers (AWS, Google, Microsoft) as well as a lot of niche providers. An overview of this is also provided in section "Business Analytics and Machine Learning as a Service (Cloud Platforms)".

Analytic Deployment

Deployment, i.e. the transfer of the application from development to productive operation, is also referred to as "DevOps". DevOps is an artificial word made up of the terms development and IT operations and refers to a methodology that relates to the interaction between development (first cycle) and operations. Development needs as much change as possible to meet the needs of changing times, while change is "the enemy" for operations. Operations require stability and thus any change is met with strong resistance. There are many basic DevOps practices. Some of them are:

Infrastructure as Code (IaC) - IaC is the practice of using the techniques, processes, and toolsets used in software development to manage the deployment and configuration of systems, applications, and middleware. Most testing and deployment failures occur when the developer environments are different from the test and production environments. Version control of these environments provides immediate benefits in terms of consistency, time savings, error rates, and audibility.

Under the practice of **continuous integration (CI),** working copies of all developer code are combined with a common master line.

Automated testing - is the practice of automatically running various tests, such as load, functional, integration, and unit tests either after you have checked in code (i.e., attached to CI) or otherwise automatically triggering one or more tests against a specific build or application.

Release Management is a practice that oversees the development, testing, deployment, and support of software releases.

Configuration Management is the practice of establishing and maintaining consistency of a product's performance with its requirements, design, and operational information throughout its life.

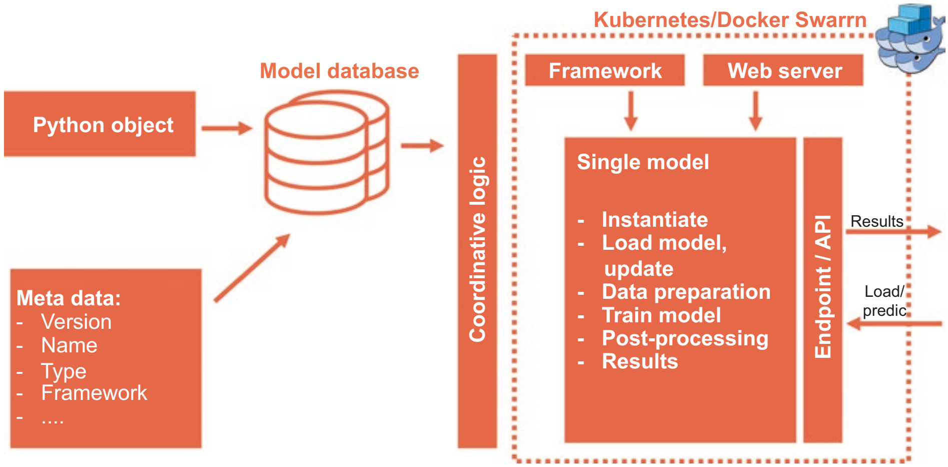


Fig. 1.4 The analytical deployment architecture with Docker Swarm or Kubernetes

Only when the models are used in production ("Analytic Deployment" or "Productive Deployment") can they create added value. Analytic deployment is, therefore, the decisive step in every business analytics **project.**

In practice, analytic deployment can be roughly divided into two procedures (see Fig. 1.4), as briefly outlined below.

Deployment within closed applications and encapsulated use in an application do not require external dependencies and are usually found on platforms or existing application landscapes, such as an SAP environment. A good example of this type of deployment is the use of the SAP HANA predictive analysis library (PAL) in the SAP environment and the SAP database platform HANA (see section "Overview of Other Microsoft Azure Services"). The PAL defines functions that can be called within SAP HANA SQLScript procedures to perform analytical algorithms. These pre-built algorithms, such as a regression (see section "Regression, Prediction, or Forecasting"), can now be called from existing applications, such as the SAP ERP system. This makes it possible to use business analytics functions with existing and native frameworks, in this case, SAP's ABAP programming language. An illustrative example is provided in the case study in section "Case Study: Analyzing Customer Sentiment in Real Time with Streaming Analytics", where a sentiment analysis for optimizing the customer experience in brick-and-mortar retail is described based on the SAP HANA database platform and the SAP customer activity repository.

Deployment as a microservice and integration into existing applications via an interface is common practice outside of closed platform systems, such as SAP. The procedure is based on the fact that the model is provided as a microservice, usually based on Python.

Here, the basic logic can be described as follows:

- The single service, optimization of a price, or automation, for example, is created with Python.
- The required logic (train, define data sources, inputs, and outputs) are created using a web server (using frameworks like Flask or FastAPI) and each process is provided by an endpoint.
- The generated objects/models are stored in a central database.
- A coordination layer, including a kind of message queue) ensures that:
  - the requests and responses are passed between the web server and the model server,
  - at a certain limit (number of requests to be processed in parallel), the container service (Docker Swarm/Kubernetes) continues to scale and create new instances,
  - in case the central database provides a newer version, this replaces the old instances.

Application Integration

Application integration is often a difficult process, especially when integrating existing legacy applications with new applications or web services. Given the vast scope of this topic, one could literally write a book about successful implementation. However, some of the basic requirements are always the same:

- Adequate connectivity between platforms.
- Business rules and data transformation logic.
- The longevity of business processes.
- The flexibility of business processes.
- Flexibility in hardware, software, and business goals.

To meet these requirements, the application environment should have a common interface for open communication, including the ability of the system to request web services and to be compatible when interfacing with other platforms and applications. The use of common software platforms (see the last chapter in this book) enables this open communication through interfaces (APIs) that underlie the paradigm of these platforms. Especially in the area of business analytics, there will rarely be a homogeneous platform for critical business applications (ERP system) and data analytics and machine learning at the same time. Rather, integration of both platforms in both directions (i.e., data access and data writing) will be necessary.

Test

Testing software describes methods for evaluating the functionality of a software program. There are many different types of software testing but the two main categories are dynamic testing and static testing. Dynamic testing is an evaluation that is performed while the program is running. In contrast, static testing is a review of the program code and associated documentation. Dynamic and static methods are often used together.

In theory, testing software is a fairly simple activity. For every input, there should be a defined and known output. Values are entered, selections or navigations are made, and the actual result is compared to the expected result. If they match, the test passes. If not, there may be an error. The point here is that until now, you always knew in advance what the expected output should be.

But this book is about a kind of software where a defined output is not always given. Importantly, in both the machine learning and analytics application examples, acceptance criteria are not expressed in terms of an error number, type, or severity. In fact, in most cases, they are expressed in terms of the statistical probability of being within a certain range.

Production/Operations

"Operate" covers the maintenance tasks and checkpoints after the rollout that enable successful deployment of the solution. This involves monitoring and controlling the applications (monitoring response times) and the hardware (server failures). This step is no different from the usual operation of other software.

The main objective of the operation is to ensure that IT services are delivered effectively and efficiently while maintaining the highest quality of service. This includes fulfilling user requests, troubleshooting service errors, resolving issues, and performing routine tasks. Some other objectives of this phase are listed below:

- Minimize the impact of service outages on daily business activities
- Ensuring access to agreed IT services only by authorized personnel and applications
- Reduction of incidents and problems
- Supporting users in the use of the service itself

Continuous Improvement

Continuous service improvement is a method of identifying and executing opportunities to improve IT processes and services and objectively measuring the impact of these efforts over time. The idea here also stems from lean manufacturing or "The Toyota Way". It was developed in manufacturing and industry to reduce errors, eliminate waste, increase productivity, optimize employee engagement, and stimulate innovation. The basic concept of continuous service improvement is rooted in the quality philosophies of twentieth-century business consultant and expert W. Edwards Deming. The so-called "Deming Circle" consists of a four-step cycle of plan, do, check, and act. This circle is executed repeatedly to achieve continuous process or service improvement.

References

1. Bracht, U.: Digitale Fabrik: Methoden und Praxisbeispiele, 2. Aufl., VDI-Buch, Geckler, D., Wenzel, S. (Hrsg.). Springer, Berlin/Heidelberg (2018)
2. Steven, M.: Industrie 4.0: Grundlagen – Teilbereiche – Perspektiven, 1. Ed., Moderne Produktion (Hrsg.). Kohlhammer, Stuttgart (2019)

3. Schneider, M.: Lean factory design: Gestaltungsprinzipien für die perfekte Produktion und Logistik. Hanser, München (2016)
4. Laney, D.: 3D data management: controlling data volume, velocity and variety. META Group Res. Note. **6**(70) (2001)
5. Gartner. Big Data. N.N. https://www.gartner.com/it-glossary/big-data/. Accessed on 22 Jan 2019 (2019)
6. Barbier, G., Liu, H.: Data mining in social media. In: Social Network Data Analytics, pp. 327–352. Springer, New York (2011)
7. Cukier, K.: Special Report: Data, Data Everywhere. (2010). February
8. Amirian, P., Lang, T., van Loggerenberg, F.: Big Data in Healthcare: Extracting Knowledge from Point-of-Care Machines. Springer, Cham (2017)
9. Gorry, G.A., Scott Morton, M.S.: A framework for management information systems. Sloan Manag. Rev. **13**, 55–70 (1971)
10. Sprague Jr., R.H.: A framework for the development of decision support systems. MIS Q. **4**, 1–26 (1980)
11. Zachman, J.A.: A framework for information systems architecture. IBM Syst. J. **26**(3), 276–292 (1987)
12. Sowa, J.F., Zachman, J.A.: Extending and formalizing the framework for information systems architecture. IBM Syst. J. **31**(3), 590–616 (1992)
13. Watson, H.J., Rainer Jr., R.K., Koh, C.E.: Executive information systems: a framework for development and a survey of current practices. MIS Q. **15**, 13–30 (1991)
14. Hosack, B., et al.: A look toward the future: decision support systems research is alive and well. J. Assoc. Inf. Syst. **13**(5), 315 (2012)
15. Watson, H.J., Wixom, B.H.: The current state of business intelligence. Computer. **40**(9), 96 (2007)
16. Eckerson, W.: BI Ecosystem of the Future. http://www.b-eye-network.com/blogs/eckerson/archives/2011/10/ (2011). Accessed on 22 Nov 2016
17. IIC: The Industrial Internet of Things Volume T3: Analytics Framework, Bd. 3. IIC, Needham, MA (2017)
18. Russom, P.: Big data analytics. TDWI Best Practices Report. Fourth Quarter. **19**(4), 1–34 (2011)
19. Watson, H., Wixom, B.: The current state of business intelligence. Computer. **40**, 96–99 (2007)
20. Watson, H.J.: Tutorial: big data analytics: concepts, technologies, and applications. Commun. Assoc. Inf. Syst. **34**(1), 65 (2014)
21. Kulkarni, R., S.I. Inc.: Transforming the data deluge into data-driven insights: analytics that drive business. In: Keynote Speech Presented at the 44th Annual Decision Sciences Institute Meeting, Baltimore (2013)
22. Awadallah, A., Graham, D.: Hadoop and the Data Warehouse: when to Use which. Copublished by Cloudera, Inc. and Teradata Corporation, California (2011)
23. Phillips-Wren, G.E., et al.: Business analytics in the context of big data: a roadmap for research. CAIS. **37**, 23 (2015)
24. Chandy, K., Schulte, W.: Event Processing: Designing IT Systems for Agile Companies. McGraw-Hill, Inc, New York (2009)
25. Watson, H.J.: Business intelligence: past, present and future, S. 153. AMCIS 2009 Proceedings, Cancun (2009)
26. Ballard, C., et al.: Information Governance Principles and Practices for a Big Data Landscape IBM Redbooks. International Business Machines Corporation, New York (2014)
27. Ponsard, C., Touzani, M., Majchrowski, A.: Combining process guidance and industrial feedback for successfully deploying big data projects. Open J. Big Data. **3**, 26–41 (2017)
28. Aggarwal, C.C.: Data Mining: the Textbook. Springer, Cham (2015)

29. Angée, S., et al.: Towards an Improved ASUM-DM Process Methodology for Cross-Disciplinary Multi-Organization Big Data & Analytics Projects. Springer, Cham (2018)
30. Veeramachaneni, K.: Why you're not getting value from your data science. https://hbr.org/2016/12/why-youre-not-getting-value-from-your-data-science (2016). Accessed on 12 May 2017
31. McKinsey: Global AI survey: AI proves its worth, but few scale impact. https://www.mckinsey.com/featured-insights/artificial-intelligence/global-ai-survey-ai-proves-its-worth-but-few-scale-impact (2019). Accessed on 21 Dec 2019

2 Artificial Intelligence

McCarthy defines artificial intelligence as "[…] the science and technology of creating intelligent machines, especially intelligent computer programs". The discipline is related to the task of using computers to understand human intelligence. Thus, many subfields and methods of AI also rely on biological patterns and processes but AI is not limited to these biologically observable methods.

As mentioned at the beginning, this book is in no way intended to delve too deeply into the individual concepts and ideas, and especially in the field of AI, this will not be necessary. Thus, it will not further discuss the concept of AI in its entirety or take up the philosophical thoughts about intelligence.[1] More so, at this point it is pointed out that we will follow a simple, yet - if properly thought through - colossal ideas of Russell and Norvig [1] on AI: AI is the science of teaching computers and machines actions that cannot yet be performed by computers and that humans are currently better at. The methods used in AI are not exclusive to this discipline.

However, the definitions and the classification are quite controversial and many boundaries are floating. A number of research sub-areas Fig. 2.1 can nevertheless be distinguished[2]:

Natural language processing (NLP) is a branch of computer science and artificial intelligence that deals with the interactions between computers and human (natural) languages, in particular, with the programming of computers to process and analyze large amounts of natural language data.

Robotics is an interdisciplinary branch of engineering and natural sciences that includes mechanical engineering, electrical engineering, information technology, computer science (and thus AI as a subfield), and others. Robotics deals with the design, construction,

[1] See the execution of Alan Turing in this regard. The mathematician and computer scientist is considered one of the most influential theorists of early computer development and computer science.

[2] However, it must be said that this classification is not uncontroversial.

F. Weber, *Artificial Intelligence for Business Analytics*,
https://doi.org/10.1007/978-3-658-37599-7_2

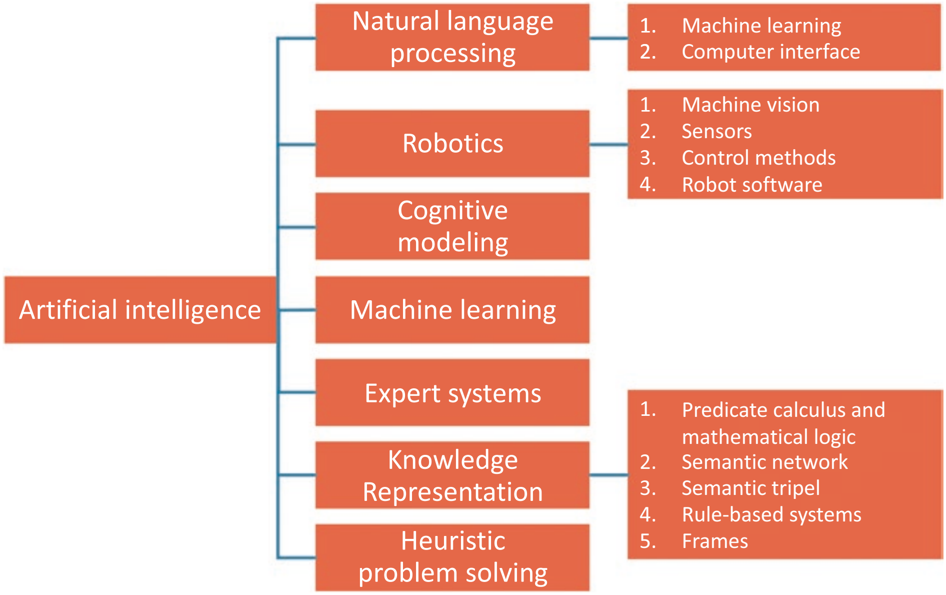


Fig. 2.1 Overview of the different branches of artificial intelligence. (representation according to Russell and Norvig [1])

operation, and use of robots as well as with computer systems for their control, sensory feedback, and information processing.

A **cognitive system** is an attempt to approximate biological cognitive processes (predominantly human) for the purpose of understanding and prediction. Cognitive models tend to focus on a single cognitive phenomenon or process (e.g., list learning), how two or more processes interact (e.g., visual search and decision making), or to make behavioral predictions for a particular task or tool (e.g., how the introduction of a new software package affects productivity).

In computer science, artificial intelligence, and mathematical optimization, a **heuristic** is a technique designed to solve a problem faster when “classical”[3] methods are too slow or to find an approximate solution when “classical” methods do not provide an exact solution. These trade-offs are also represented as a triangle of goals between optimality, completeness (also called accuracy or precision), and solution time (speed).

Knowledge representation and **logic** is a field devoted to representing information about the environment in a form that a computer system can use to solve complex tasks such as diagnosing a medical condition or dialoguing in a natural language. Knowledge representation incorporates insights from psychology about how people solve problems and represent knowledge to design formalisms so that complex systems are easier to design and build. Knowledge representation and reasoning also incorporate insights from

[3] Mostly mathematical optimization systems are meant here.

logic to automate various types of reasoning, such as the application of rules or the relationships of sets and subsets.

In the field of business analytics, one area of AI stands out is **machine learning (ML)**. ML is a discipline within AI research that focuses on improving learning based on data. Ultimately, it is about the extent to which tasks are continuously better solved by the machine through particularly good training data or particularly large amounts of data from algorithms.

Machine Learning

Machine learning is an essential part of AI and is so popular that it is sometimes confused with artificial intelligence (at least the two terms are often used interchangeably).

The algorithms used in machine learning can be broadly divided into three categories (see Fig. 2.2 and the comparison in section "Unsupervised Learning"): supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves feedback to indicate when a prediction is correct or incorrect, while unsupervised learning involves no response: the algorithm simply attempts to categorize data based on its hidden structure. Reinforcement learning is similar to supervised learning in that it receives feedback but not necessarily for every input or state. Below, we will explore the ideas behind the learning models and present some key algorithms used for each of them. Machine learning algorithms are constantly changing and evolving. However, in most cases, the algorithms tend to integrate into one of three learning models. The models exist to automatically adapt in some way to improve their operation or behavior. In supervised learning, a dataset contains its desired outputs (or labels) so that a function can compute an error for a given prediction. Supervision occurs when a prediction is made and an error (actual vs. desired)

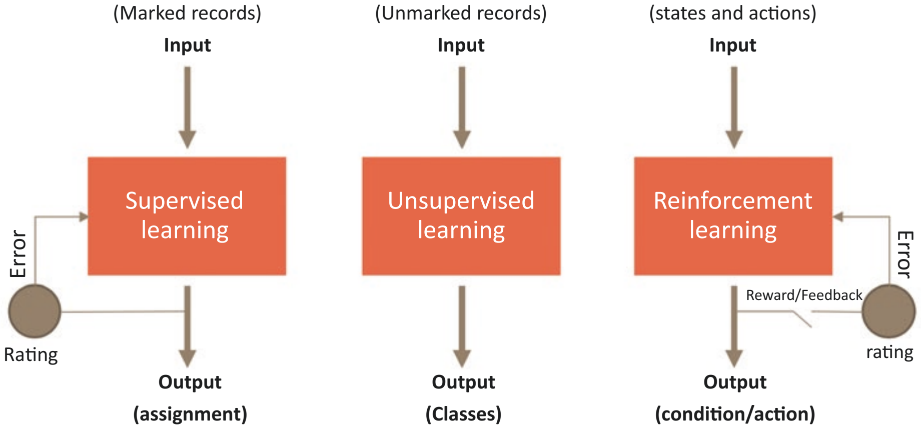


Fig. 2.2 Different types of machine learning

is generated to change the function and learn the mapping. In unsupervised learning, the data set does not contain the desired output, so there is no way to monitor the function. Instead, the function attempts to segment the dataset into "classes" so that each class contains a portion of the dataset with common features. Finally, in reinforcement learning, the algorithm attempts to learn actions for a given set of states that lead to a target state. An error is not issued after each example (as in supervised learning) but rather when a reinforcement signal is received (e.g., the target state is reached). This behavior is similar to human learning, where feedback is not necessarily given for all actions but only when a reward is warranted.

Supervised Learning

Supervised learning is the simplest of the learning models.[4] Learning in the supervised model involves creating a function that is trained using a training data set and can then be applied to new data. Here, the training dataset contains labeled records (labels) so that the mapping to the desired result given the set input is known in advance. The goal is to build the function in such a way that beyond the initial data a generalization of the function is possible, i.e. to assign unknown data to the correct result.

In the first phase, one divides a data set into two types of samples: training data and test data. Both training and test data contain a test vector (the inputs) and one or more known desired output values. The mapping function learns with the training dataset until it reaches a certain level of performance (a metric of how accurately the mapping function maps the training data to the associated desired output). In supervised learning, this is done for each training sample by using this error (actual vs. desired output) to adjust the mapping function. In the next phase, the trained mapping function is tested against the test data. The test data represents data that has not been used for training and whose mapping (desired output) is known. This makes it very easy to determine a good measure of how well the mapping function can generalize to new and unknown data [3].

To tackle a given problem on a generic basis of supervised learning, several steps need to be taken [3]:

1. Identification of different training examples. Thus, a single handwritten character, word, or complete sentence can be used for handwriting analysis.
2. The second step is to assemble a training set that must represent the practical application of a function. Therefore, a set of input objects and equivalent results must be determined either from measurements or from experts. In the example, to determine the transfer of the image as a matrix of black (where there is writing) and white (where there is no writing) fields to a mathematical vector.

[4] Further information can be found in [2].

3. The third step is to identify a particular input object (character) that would represent a learned function. The accuracy of a learned function depends heavily on the representation of the input object. Here, the input object is converted into a feature vector consisting of multiple features describing an object. The total number of features should be small due to the curse of dimensionality[5] but must contain enough information to accurately predict the output.
4. The fourth step is to identify the structure of the learned function along with the corresponding learning algorithm.
5. The learning algorithm is now executed on the cumulative training set. Few of the supervised learning algorithms require users to evaluate the control parameters to be adjusted by performance optimization on a subset called the validation set.
6. The final step is to assess the accuracy of the learned function. After the processes of learning and parameter setting, the performance of the function must be measured on the test set, which is different from the original training set.

Unsupervised Learning

Unsupervised learning is also a relatively simple learning model.[6] However, as the name implies, it lacks a supervisory authority and there is no way to measure the quality of the results. The goal is to build a mapping function that categorizes the data into classes based on features hidden in the data.

As in supervised learning, one uses two phases in unsupervised learning. In the first phase, the mapping function segments a dataset into classes. Each input vector becomes part of a class but the algorithm cannot assign labels to these classes.

This data is not labeled (unlike supervised learning where the labeling is done by the user in advance), which shows that the input variables (X) do not have equivalent output variables. Here, the algorithms have to identify the data structures themselves [7]. Unsupervised learning can again be simplified into two different categories of algorithms:

- Clustering (section "Types of Problems in Artificial Intelligence and Their Algorithms"): The problem to be solved with these algorithms occurs when trying to identify the

[5] The curse of dimensionality refers to several phenomena that occur when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional environments, such as the three-dimensional physical space of everyday experience. The common theme of these problems is that as the dimensionality increases, the volume of the space increases so rapidly that the available data becomes sparse. In machine learning problems where learning is done from a limited number of data samples in a high dimensional feature space, where each feature has a set of possible values, an enormous amount of training data is usually required to ensure that there are multiple samples with each combination of values. More information on the curse and possible solutions can be found, for example, in [4].

[6] For more detailed information on this learning model, see [5] or [6].

integral groupings of the data, such as grouping customers based on their buying behavior.
- Association (section "Types of Problems in Artificial Intelligence and Their Algorithms"): The problem to be solved with these algorithms arises when trying to find the rules to describe a large part of the available data, such as people who tend to buy both product X and Y. The problem to be solved with these algorithms arises when trying to find the rules to describe a large part of the available data.

Excursus: Semi-Supervised Learning

In the previous two types, there is either no pre-known label (labels) of the dataset or there are labels for all observations. Semi-supervised learning is positioned exactly between these two extremes. Namely, in many practical situations, the cost of labeling is quite high, as it requires skilled human professionals. In the absence of labeling in the majority of observations (but availability in a smaller subset), semi-supervised algorithms are the best candidates. These methods rely on the idea that, although the group membership of the unmarked data is unknown, these data contain important information about the group parameters and for generalization.

Reinforcement Learning

Reinforcement learning is a learning model with the ability to learn not only how to map an input to output but also how to map a set of inputs to outputs with dependencies (e.g., Markov decision processes).[7] Reinforcement learning exists in the context of states in an environment and the possible actions in a given state. During the learning process, the algorithm randomly explores the state-action pairs within an environment (to build a state-action pair table), then in practice exploits the rewards of the state-action pairs to select the best action for a given state that leads to a target state.

In this context, reinforcement learning is mostly implemented by (partially) autonomous software programs, so-called agents. These agents interact with the environment through discrete time steps. The agent's ultimate goal is to maximize rewards. At a given time t, the agent receives an observation and the maximum possible reward [10]. Action is now selected from the available set of actions, which are then sent to the affected environment. Thus, a new state is found and the associated reward is determined with this transition. The reinforcement can be either positive or negative. It is the occurrence of an event resulting from a particular behavior that increases the frequency and strength of the behavior. In this context, an optimally acting agent must be able to consider the long-term effects of its actions, even if the immediate reward is negative [11]. Therefore, reinforcement learning is suitable for topics such as short- and long-term reward trade-offs. The use of

[7] For further information see [8] or [9].

functional approximations in larger settings and the use of examples to optimize performance are the key elements that enhance reinforcement learning. The situations in which reinforcement learning is used are characterized by the absence of an analytical situation, but the environmental model is known. Thus, the simulation model for the environment is known and the information about the environment can be gathered by interacting with it [12]. The first two issues can be classified as planning problems and the last one is a true learning problem.

To create intelligent programs (the agents), reinforcement learning generally goes through the following steps:

1. The input state is monitored by the agent.
2. The decision function is used to make the agent perform an action.
3. After performing the action, the agent receives a reward or reinforcement (positive or negative) from the environment.
4. The information (state action) about the reward is stored.

Overview of the Types of Machine Learning

Table 2.1 shows an overview of different types of machine learning.

Neural Networks

Since neural networks are the basis of most innovations in the field of artificial intelligence and machine learning (self-driving cars, chatbots like Siri, etc.), they are explained in a bit more detail below.

Table 2.1 Different types of machine learning

Type of machine learning	When to use?	Possible algorithms and methods
Supervised learning	When it is known how the input data is classified and what type of behavior is to be predicted	Regression, decision tree, naive Bayes, vector machines, random forest, neural network
Unsupervised learning	When it is known how the input data will be classified and patterns in the data should be detected	K-means, recommender system, hierarchical clustering
Reinforcement learning	When there is little training data and the ideal target state cannot be clearly defined or must first be developed through interaction with the environment	Q-learning, temporal difference (TD), deep adversarial networks

The three types of machine learning are each used in different situations and each involves different algorithms. Selected problems and algorithms can be found in section “Reinforcement Learning”

Neural networks are a set of algorithms loosely designed after the human brain and fundamentally designed to recognize patterns. They interpret sensory data through a form of machine perception, labeling, or clustering of raw data. The patterns they recognize are numerical, contained in vectors into which all real-world data, be it images, sound, text, or time series, must be translated.

The basic building blocks of neural networks are neurons. These form the smallest basic unit of a neural network. A neuron takes input, computes with it, and generates an output. This is what a 2-input neuron looks like (see Fig. 2.3).

Three things happen to each neuron. First, each input is multiplied by a weight w:

$$x1 \to x1 * w1$$
$$x2 \to x2 * w2$$

Then, all weighted inputs are added with a bias b "bias":

$$(x1 * w1) + (x2 * w2) + b$$

Finally, the sum is passed through an activation function:

$$y = f(x1 * w1 + x2 * w2 + b)$$

The activation function is used to turn an unbounded input into an output that has a predictable shape. A commonly used activation function is the sigmoid function, see Fig. 2.4.

The sigmoid function only returns numbers in the range of (0,1). You can think of this as compression: (−∞,+∞) becomes (0,1) - large negative numbers become 0 and large positive numbers become 1.

If the activation function now results in 1, the neuron is considered to be activated and "fires", i.e. it passes on its value. This is because a neural network is nothing more than a

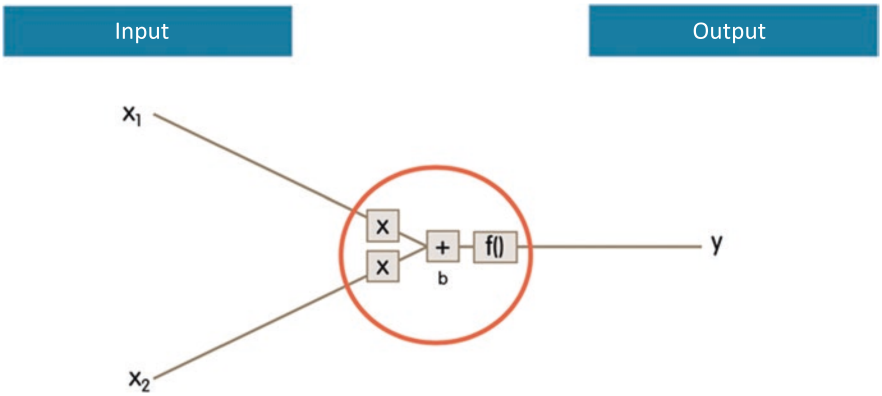
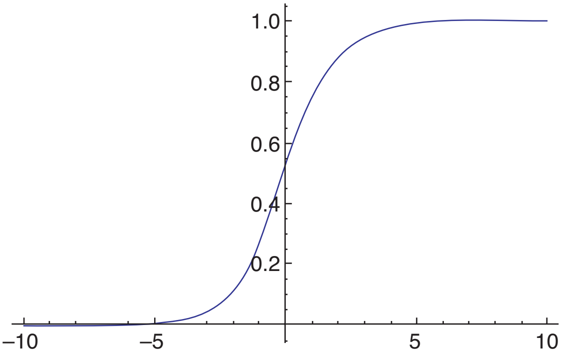


Fig. 2.3 Representation of a neuron and the underlying mathematical processes

Fig. 2.4 Sigmoid function



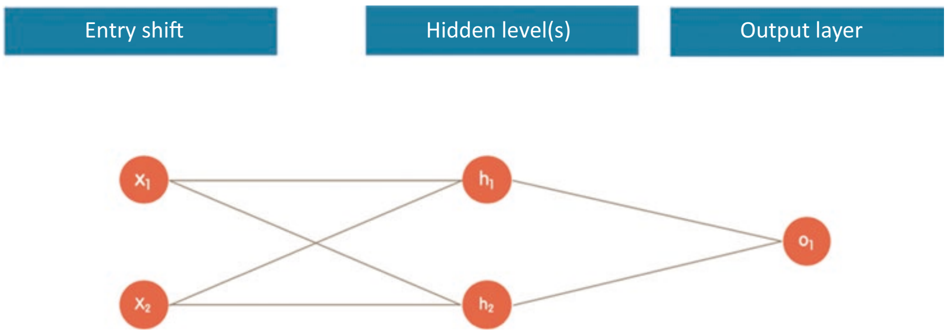


Fig. 2.5 Minimal neural network with one hidden layer

set of neurons that are connected to each other. This is what a simple neural network might look like, Fig. 2.5:

This network has two inputs, a hidden layer with two neurons (*h1* and *h2*) and an output layer with one neuron (*o1*). A hidden layer is any layer between the input (first) layer and the output (last) layer. In most practical cases, there will be several hundreds of hidden layers!

Crucially, the inputs to o1 are the outputs of h1 and h2 - this is precisely what makes loose neurons a neural network.

The neural network is now used in two ways. During learning (training) or normal use (after training has taken place), patterns of information are fed into the network via the input layer, triggering the layers of hidden units, which in turn reach the output units. This interconnected design is called a feedforward network.

Each neuron receives input signals from the neurons on the left (figuratively) and the inputs are multiplied by the weights of the connections. Each unit sums all the inputs received in this way (for the simplest type of network) and, when the sum exceeds a certain threshold (value of the activation function), the neuron "fires" and triggers the following neurons (on the right).

For a neural network to learn, some kind of feedback must be involved. In a figurative way, neural networks learn in much the same way as small children learn by being told what they did right or wrong.

Neural networks learn in a similar way, typically through a feedback process called backpropagation. In this process, the output of the network is compared to the output it should produce for a correct result. The discrepancy (difference) between the two states is used to change the weights (w) of the connections between the units in the network, working from the output units to the hidden units to the input units-that is, backward (hence the word backpropagation). Over time, backpropagation causes the network to adapt (learn) and reduce the difference between actual and intended output to the point where the two match exactly, so that the network computes things exactly as expected.

Once the network has been trained with enough learning examples, it reaches a point where it can be used with a completely new set of inputs. This is because the neural network now allows the generalization of the results learned from the learning phase and applies to new situations (data).

Types of Problems in Artificial Intelligence and Their Algorithms

Classification

Classification is the process of predicting the class of given data points. Classes are sometimes referred to as targets/labels or categories. Predictive modeling of classification is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

For example, spam detection in email service providers can be identified as a classification problem. This is a binary classification because there are only 2 classes: spam (1) and regular emails (0). A classifier uses training data to understand how certain input variables relate to the class. In this case, known spam and non-spam emails must be used as training data. If the classifier is trained accurately, it can be used to detect an unknown email.

Classification belongs to the category of supervised learning where the targets are provided along with the input data. There are applications of classification in many domains, such as lending, medical diagnosis, target marketing, etc. There are two types of learners in classification as lazy and eager.

- **Lazy learners -** simply store the training data and wait for test data to appear. In this case, classification is performed based on the most frequently used data in the stored training data. Compared to eager learners, lazy learners have less training time but more time for a prediction. Example algorithms would be k-nearest-neighbor (k-NN) or case-based reasoning (CBR).
- **Eager learners -** construct a classification model based on the given training data before receiving data for classification. The system must be able to settle on a single

hypothesis that covers the entire instance space. Due to the model construction, eager learners need a lot of time for training and less time for a prediction. Example algorithms would be decision trees, Naive Bayes, artificial neural networks.

There are many classification algorithms currently available but it is not possible to determine which one is superior to the other. It depends on the application and the type of dataset available. For example, if the classes are linearly separable, the linear classifiers such as logistic regression or Fisher's discriminant functions may outperform sophisticated models and vice versa.

Decision tree - builds classification or regression models in the form of a tree structure (Fig. 2.6). It uses an if-then rule set that is mutually exclusive and complete for classification. The rules are learned sequentially using the training data one by one. Each time a rule is learned, the tuples covered by the rules are removed. This process continues on the training set until a termination condition is satisfied. Decision trees generally follow a recursive top-down principle in structure. It is a top-down principle (from top to bottom) since a run always starts from the tree root and continues from there successively into two new branches down the tree (towards the treetop).

All attributes should be categorical. Otherwise, they should be discretized in advance. Attributes at the top of the tree have a greater impact on classification and are identified using the concept of information gain. A decision tree can easily be overbuilt and create too many branches and may reflect anomalies due to noise or outliers. An overfitted model will have very poor performance on the unseen data, although it will give an impressive performance on the training data. This can be avoided by pruning, which stops the tree construction early, or by "re-pruning", which removes branches from the mature tree.

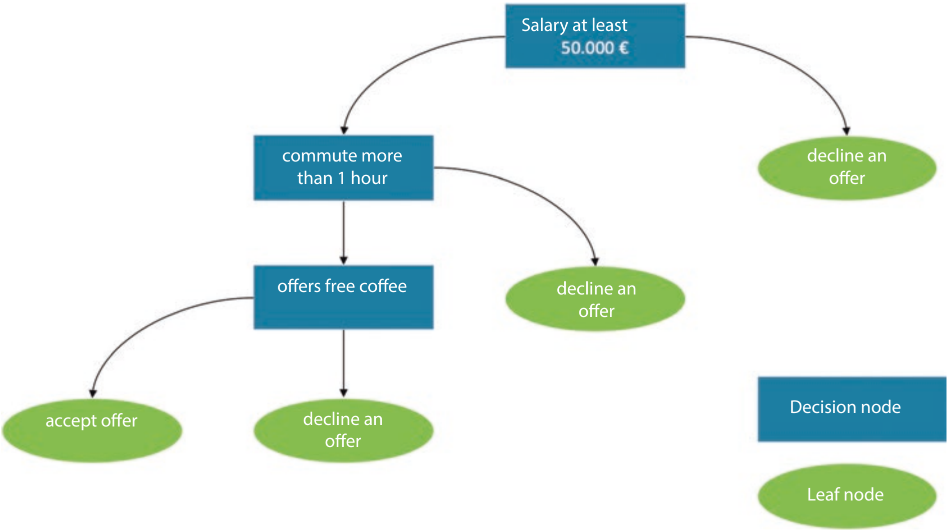


Fig. 2.6 Visualization of a decision tree

The **Naive Bayes**-classifier is a probabilistic classifier inspired by the Bayes theorem from the simple assumption that the attributes are conditionally independent. The classification is done by deriving the maximum posterior, i.e., the maximum P(Ci|X) with the above assumption for the Bayes theorem. This assumption significantly reduces the computational cost by counting only the class distribution. Even though the assumption does not hold in most cases since the attributes are dependent, the naive Bayes classifier was surprisingly able to perform impressively. This classifier is a very simple algorithm and good results were obtained in most cases. It can be easily scaled to larger datasets as it takes linear time, rather than performing an expensive iterative approximation as used in many other classifiers. Naive Bayes algorithms can suffer from a problem called the "zero likelihood problem". If the conditional probability for a given attribute is zero, there can be no valid prediction. This must be explicitly fixed with a Laplace approximation.

Artificial Neural Networks (ARNs) are a set of connected input/output units where each connection is assigned a weight (Fig. 2.7). The model was developed by psychologists and neurobiologists to obtain and test computational analogs of neurons. During the learning phase, the network learns by adjusting the weights to predict the correct class label of the input tuples. There are many network architectures available now like feed-forward, convolutional, recurrent, etc. The appropriate architecture depends on the application of the model. In most cases, feed-forward models give relatively accurate results, and convolutional networks perform better especially for image processing applications. Depending on the complexity of the function to be represented by the model, there may be several hidden layers in the model. More hidden layers make it possible to model complex relationships, such as deep neural networks. However, if there are many hidden layers, it takes a lot of time to train and adjust the weights. The other drawback is the poor interpretability of the model compared to other models, such as decision trees due to the unknown symbolic meaning behind the learned weights. Artificial neural networks have nevertheless shown impressive performance in most real-world applications. A

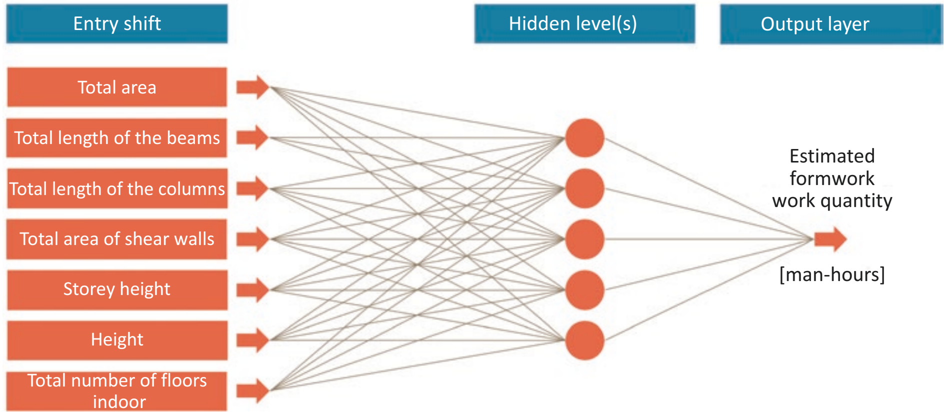

Entry shift
Hidden level(s)
Output layer
Total area
Total length of the beams
Total length of the columns
Total area of shear walls
Storey height
Height
Total number of floors indoor
Estimated formwork work quantity
[man-hours]

Fig. 2.7 Visualization of the artificial neural networks

neural network has a high tolerance to noisy data and can classify untrained patterns. In general, artificial neural networks with continuous-valued inputs and outputs perform better.

K-nearest-neighbor (KNN) is a lazy learning algorithm that stores that all instances correspond to training data points in n-dimensional space (see Fig. 2.8). When unknown discrete data is received, it analyzes the nearest k number of stored instances (nearest neighbors) and returns the most common class as a prediction, and for real data, it returns the mean of the k nearest neighbors. In the distance weighted nearest neighbor algorithm, it weights the contribution of each of the k neighbors according to their distance with the following query giving more weight to the nearest neighbors. Normally, KNN is robust to noisy data as it averages the nearest neighbors in each case.

Dependencies and Associations

Association analysis, as the name implies, is a form of analysis that looks for associations between objects. It is also called affinity analysis, and a particular subset of this form of analysis is often colloquially referred to as market basket analysis, as this is the most commonly used and discussed application. In particular, the scope of association analysis is the aspect of shopping cart analysis, i.e., “What products are bought together in a shopping cart?” The results of association analysis usually take the form of rules, such as “If item A is purchased, then so is item B.” The quality or usefulness of these rules is evaluated by calculating the number of baskets that support the rule, i.e. where the combination exists, divided by the total number of baskets. This metric or statistic is called rule support. It is then calculated how good the rule is at predicting the “right side” of the

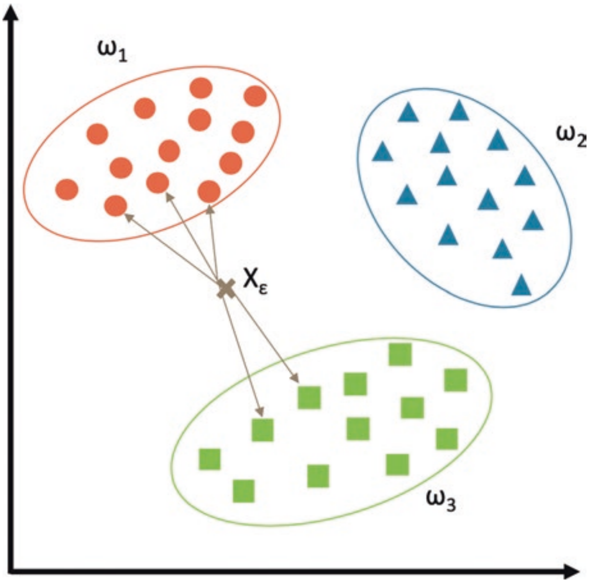


Fig. 2.8 Visualization of the k-nearest-neighbor algorithm

rule, point B in our example, given the "left side" of the rule, point A in our example. This measure or statistic is the number of baskets in which A and B are present, divided by the number of baskets with A in them, expressed as a percentage. It is referred to as "rule support". Finally, we compute a measure "lift", which is defined as the confidence of the combination of items divided by the support of the outcome. "Uplift" is the ratio of buyers of A and B to buyers of B alone. It measures how much better the rule is than just guessing the "right side" of the rule. If it is greater than 1, then B is more often bought with A; if it is less than 1, then B is more often bought alone, in which case the rule is not very good. The calculations are very simple but in practice, the challenge of association analysis is usually the very large amount of transaction data and thus the performance of the hardware. This challenge can be addressed by limiting the number of rules to be extracted. Interpreting the results is not always as straightforward as it seems, especially when rules produce either trivial or seemingly nonsensical associations.

The main strength of association analysis is that it provides clear results. In addition, the calculations are simple and therefore easy to understand, which increases the chance that management will implement the results. The rules provided are actionable for many applications, especially to suggest cross-selling and up-selling opportunities for products. These are great advantages in predictive analytics, where algorithms are often very complex and thus difficult to explain, which raises doubts among respondents who are asked to implement their results in business processes. One of the biggest weaknesses of the Apriori algorithm is that it requires exponentially more computations as the amount of data grows.

For this reason, there are several other shortcuts to the algorithm, such as Apriori Lite.[8] Some of the results may be trivial and thus worthless. Conversely, some of the results may be inexplicable - but this latter problem applies to much of predictive analysis (mathematical relationships can be found but is the relationship mathematical or random?). Another weakness is that the algorithm discounts rare items since it is essentially looking for common items. The weaknesses are far outweighed by the strengths. Although data sets can be huge, in practice they are often broken down into segments, as users typically search for relationships within product hierarchies or by location, such as retail stores, and thus for smaller data sets.

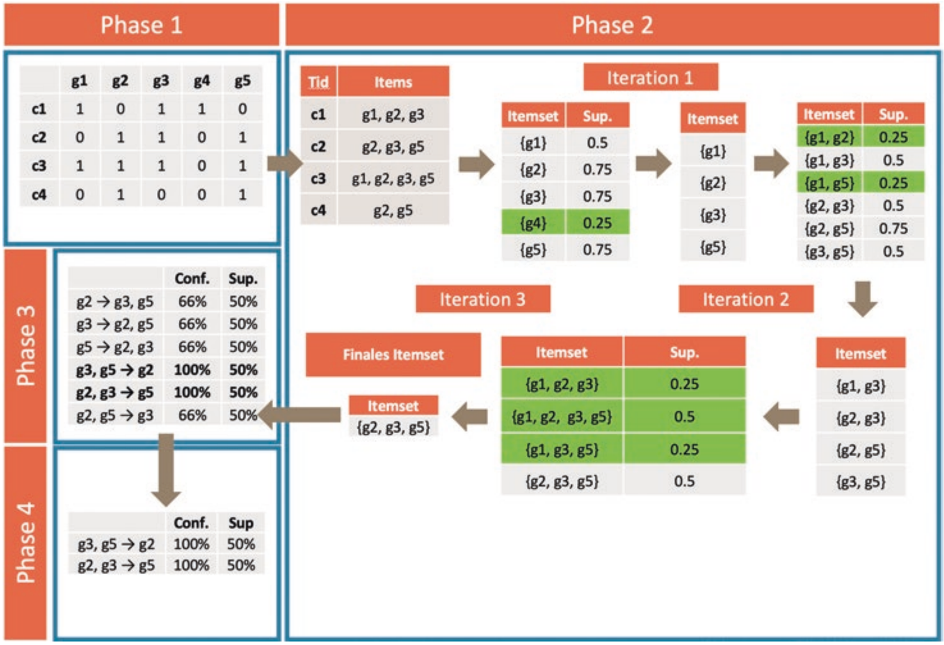
The **Apriori algorithm** is an example from association analysis, which was developed specifically for use in very large data sets [13]. In the context of association analysis, rules are to be generated (see Fig. 2.10) "which describe the relationships between the elements (items) occurring in the data sets of a dataset" [14]. The starting point of the Apriori algorithm is a dataset D of individual transactions, which in turn consist of a set of items, where items represent uninterpreted and discrete entities [15]. Here, an association rule X → Y consists of an item set X as a premise and the item set Y as a conclusion, where X and Y must be disjoint. A transaction t satisfies a rule X → Y if all the items in the rule are also in the transaction (thus, $(X \cup Y) \subseteq t$ holds). Here, the rules are evaluated using two

[8] A survey of the various developments around the Apriori algorithm can be found in [13] or [14].

$$support(X \rightarrow Y) = \frac{|\{t \in \mathcal{D} \mid (X \cup Y) \subseteq t\}|}{|\mathcal{D}|}$$

$$confidence(X \rightarrow Y) = \frac{|\{t \in \mathcal{D} \mid (X \cup Y) \subseteq t\}|}{|\{t \in \mathcal{D} \mid X \subseteq t\}|} = \frac{support(X \rightarrow Y)}{support\,(X)}$$

Fig. 2.9 Formal representation of the support and confidence of the Apriori algorithm



	g1	g2	g3	g4	g5
c1	1	0	1	1	0
c2	0	1	1	0	1
c3	1	1	1	0	1
c4	0	1	0	0	1

Tid	Items
c1	g1, g2, g3
c2	g2, g3, g5
c3	g1, g2, g3, g5
c4	g2, g5

Itemset	Sup.
{g1}	0.5
{g2}	0.75
{g3}	0.75
{g4}	0.25
{g5}	0.75

Itemset
{g1}
{g2}
{g3}
{g5}

Itemset	Sup.
{g1, g2}	0.25
{g1, g3}	0.5
{g1, g5}	0.25
{g2, g3}	0.5
{g2, g5}	0.75
{g3, g5}	0.5

Itemset
{g1, g3}
{g2, g3}
{g2, g5}
{g3, g5}

Itemset	Sup.
{g1, g2, g3}	0.25
{g1, g2, g3, g5}	0.5
{g1, g3, g5}	0.25
{g2, g3, g5}	0.5

Finales Itemset

Itemset
{g2, g3, g5}

	Conf.	Sup.
g2 → g3, g5	66%	50%
g3 → g2, g5	66%	50%
g5 → g2, g3	66%	50%
g3, g5 → g2	**100%**	**50%**
g2, g3 → g5	**100%**	**50%**
g2, g5 → g3	66%	50%

	Conf.	Sup
g3, g5 → g2	100%	50%
g2, g3 → g5	100%	50%

Fig. 2.10 Visualization of the Apriori algorithm

probabilistic measures, support and confidence. The algorithm ignores rules that lie below a minimum value of these two values to be defined in advance.

The support represents the probability that an item set occurs in a transaction. It is, therefore, the relative frequency with which the rule occurs in the database. It is relatively unlikely that a rule is valid for all transactions. Therefore, confidence is defined as the proportion of transactions that satisfy the premise and the conclusion. Formally, this holds:

The Apriori algorithm contains two successive steps (see Fig. 2.9). First, all item sets are calculated whose support lies within the previously defined interval. In the second step, the support value is determined for these items, also referred to as frequent items (Fig. 2.10).

The advantages of this algorithm are that it is specifically designed for use in large datasets and is, therefore, less resource-intensive and faster than similar association analysis algorithms.[9] On the other hand, there is no need to pre-define any constraints on the algorithm, other than subjectively defining the relevant intervals of support and confidence - which is why even trivial or uninteresting rules are included in the result [16].

Clustering

Cluster analysis is concerned with organizing data into groups with similar characteristics. Ideally, the data within a group are closely matched, while the groups themselves are very different. In other words, the object distances between the clusters are within the cluster ("inter-cluster") but at the same time, the distances between the clusters ("intra-cluster") are large.

Market segmentation is one of the main applications of cluster analysis. Rather than marketing generically to everyone, there is a consensus that it is more beneficial to focus on specific segments, such as with targeted product offerings. There is an entire industry devoted to market segmentation. Segmentation has been used to find groups of similar customers to select test markets for promotional offers, to try to understand the key attributes of the segments, and to track the movement of customers from different segments over time to understand the dynamics of customer behavior.

We have seen how cluster analysis can be used to refine predictive analysis when dealing with large and complex data sets. A parallel example of this would be that a company has thousands of products or hundreds of stores, and strategies are to be developed to manage these products and stores. This is not to create a hundred or even a thousand strategies, so the products and stores need to be grouped and a manageable number of strategies developed. Where each strategy then applies only to groups of products or stores. An unusual example of cluster analysis was that of the US Army who wanted to reduce the number of different unit sizes and so analyzed many measurements of body size and derived a size system where individuals were assigned to particular size groups/clusters.

Cluster analysis is probably the most widely used class of predictive analytic methods with applications in a whole range of fields, such as crime pattern analysis, medical research, education, archaeology, astronomy, or industry. Clustering is indeed ubiquitous.

K-means clustering - is the most famous clustering algorithm. The reasons for this are obvious: it is easy to understand and implement. The diagram below (see Fig. 2.11) serves as an illustration. First, a set of groups or classes are selected to be used and initialized randomly according to their corresponding midpoints. To determine the number of classes to use, one should briefly look at the data and then try to identify each grouping [17]. The midpoints are vectors of equal length since all data point vectors are X-coordinates. Each data point is categorized by calculating the distance between that point and the center of each predefined group and then assigning the point to the closest group.

[9] See [14] or [16].

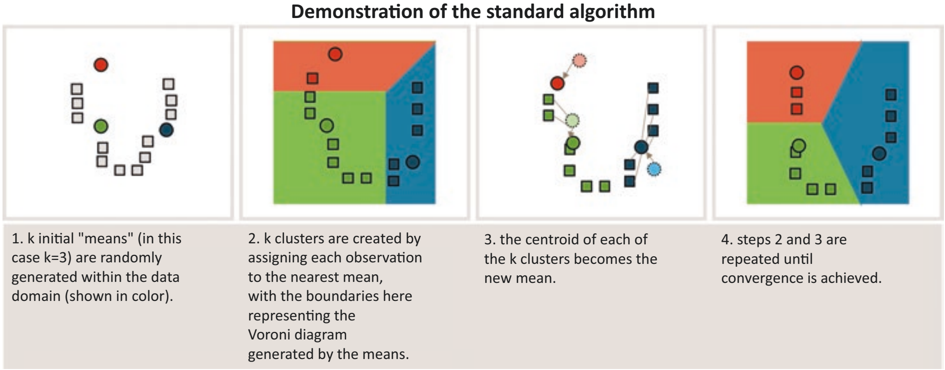

Demonstration of the standard algorithm
1. k initial "means" (in this case k=3) are randomly generated within the data domain (shown in color).
2. k clusters are created by assigning each observation to the nearest mean, with the boundaries here representing the Voroni diagram generated by the means.
3. the centroid of each of the k clusters becomes the new mean.
4. steps 2 and 3 are repeated until convergence is achieved.

Fig. 2.11 Steps of K-means clustering

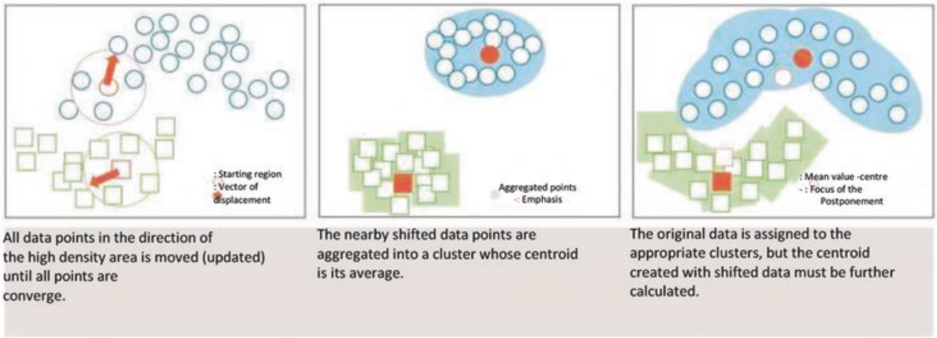

: Starting region
: Vector of displacement
Aggregated points
-: Emphasis
: Mean value -centre
-: Focus of the Postponement
All data points in the direction of the high density area is moved (updated) until all points are converge.
The nearby shifted data points are aggregated into a cluster whose centroid is its average.
The original data is assigned to the appropriate clusters, but the centroid created with shifted data must be further calculated.

Fig. 2.12 Visualization of the mean-shift algorithm according to [18]

Based on these classified points, the group center can be recalculated using the mean of all vectors. These steps are repeated for a fixed number of iterations or until the centers show very little change between successive iterations. The group centers can also be randomly initialized several times and then the run with the best results is selected. K-means is very fast because it only computes the distances between the group centers and it requires very few computational operations. So its linear complexity is O(n) [17]. K-means also has some disadvantages. First, one has to "randomly" choose how many groups/classes there are. Also, this causes different runs of the algorithm to produce different clustering results as a result. In this way, the results may not be stable and may lead to confusion.

Mean shift clustering - is a clustering algorithm that iteratively assigns data points to clusters by shifting points in the direction of the mode (see graphical representation in Fig. 2.12). The mode can be understood as the highest density of data points (in the region, in the context of the mean-shift). Therefore, it is also referred to as the mode search algorithm. The mean-shift algorithm has applications in the field of image processing and computer vision. Given a set of data points, the algorithm iteratively assigns each data point to the nearest cluster centroid. The direction to the next cluster centroid is determined by where most of the nearby points are located. Thus, at each iteration, each data point moves closer to where most of the points are located, which leads or will lead to the cluster centroid. When the algorithm stops, each point is assigned to a cluster.

Unlike the popular K-means algorithm, mean shifting does not require prior specification of the number of clusters. The number of clusters is determined by the algorithm with respect to the data.

Density-based clustering - is basically based on the mean shift algorithm but has some advantages. This type of clustering starts with any data point that is not visited. The neighborhood of the point is extracted with a distance, all points within the distance are thus neighborhood points. If we have determined a sufficient number of points in the neighborhood, the clustering process begins and the current data point is considered the first point in the new cluster. If not, it is called noise, which may later become part of the cluster. In both cases, the point is labeled as "visited" [19]. With the first point of the new cluster, the points lying in it, the neighborhood is also part of this cluster. The process of rendering all points in the neighborhood to the same cluster is repeated for each new point added to the cluster group.

Regression, Prediction, or Forecasting

Many applications involve modeling relationships between one or more independent variables and a dependent variable. For example, the amount of revenue or sales can be predicted based on the price set or projected into the future as a trend. As other examples, a company might predict sales based on gross domestic product, or a marketing researcher might predict the odds that a particular model of the automobile will be purchased based on a survey of consumer attitudes toward the car brand and income levels. Regression analyses are tools to build such models and predict future outcomes [20]. The main focus here is to get a basic understanding of what the data say and how to interpret them through trend lines, regression models, and statistical questions and how they are related. Understanding both the mathematics and the descriptive properties of various functional relationships is important for building predictive analytic models. One often begins by constructing a diagram of the data to understand the dependencies in terms of content and to select the appropriate type of functional relationship for the analytic model. For cross-sectional data, it is best to use a scatter plot; for time series or data series, it is best to use a line plot.

The most common types of mathematical functions used in predictive analytic models are the following:

- **Linear function y = a + bx** - Linear functions show a continuous increase or decrease over the range of x. This is the simplest type of function used in predictive models. It is easy to understand and can approximate behavior quite well over small ranges of values.

- **Logarithmic function y = ln(x)** - Logarithmic functions are used when the rate of change of a variable increase or decreases rapidly and then levels out, such as when returns to scale decrease. Logarithmic functions are often used in marketing models where constant percentage increases in advertising, for example, lead to constant absolute increases in sales.
- **Polynomial function $\mathbf{y = ax^2 + bx + c}$** (second-order quadratic function) or $\mathbf{y = ax^3 + bx^2 + dx + e}$ (third-order cubic function) and so on. A second-order polynomial is parabolic and has only one hill or valley; a third-order polynomial has one or two hills or valleys. Sales and revenue models or price elasticity are often polynomial functions.
- **Power function y = axb** - Power functions define phenomena that increase at a certain rate. Learning curves that express improvement times in the execution of a task are often modeled with power functions with $a > 0$ and $b < 0$.
- **Exponential function y = abx** - Exponential functions have the property of increasing or decreasing at ever-increasing rates. For example, the perceived brightness of a light bulb increases at a decreasing rate as the power increases. In this case, a would be a positive number, and b would be between 0 and 1. The exponential function is often defined as y = abx, where b = e (approximately 2.71828).

Regression analysis is a tool for building mathematical and statistical models that characterize relationships between a dependent variable (which must be a ratio variable and is not categorical) and one or more independent or explanatory variables, all of which are numerical (but can be either ratio or categorical).

Two broad categories of regression models are commonly used in the business environment: (1) regression models of cross-sectional data and (2) regression models of time series data, where the independent variables are time or a function of time and the focus is on predicting the future. A regression model that includes a single independent variable is called simple regression. A regression model that includes two or more independent variables is called multiple regression.

Least squares regression - If the data has a linear relationship between variables X and Y, the task is to find a straight line that best describes that linear relationship. The resulting function can then be used to calculate the output for any given input.

This straight line/line is called the regression line and has the equation $\hat{y} = a + b x$. The least squares regression line is the line that makes the vertical distance from the data points to the regression line as small as possible. It is called "least squares" because the best fit line is one that minimizes the variance (the sum of the squares of the errors). This can be a bit hard to visualize but the main point is that you want to find the equation that fits the points as closely as possible.

Advantages: Least squares allow residuals to be treated as a continuous quantity in which derivatives (measuring how much the output of a function changes when an input changes) can be found. This is invaluable since the point of finding an equation in the first place is to predict where other points might lie on the line (including points far beyond the original points).

Disadvantages: outliers can have a disproportionate effect when using the least squares fitting method to find an equation for a curve. This is because the squares of the offsets are used instead of the absolute value of the offsets; outliers naturally have larger offsets and affect the line more than points closer to the line. These disproportionate values can be beneficial in some cases.

Multiple regression analysis - Ordinary linear regression is usually insufficient to account for all the real factors that affect an outcome. To illustrate this point:

- Simple regression is performed on a function of the construction: $Y = b_0 + b_1\ x$
- Multiple regression is performed on a function of the structure: $Y = b_0 + b_1\ x1 + b_0 + b_1\ x2\ ...\ b_0\ ...\ b_1\ x_n$

For example, Fig. 2.13 contrasts a single variable (number of physicians) with another variable (life expectancy of women).

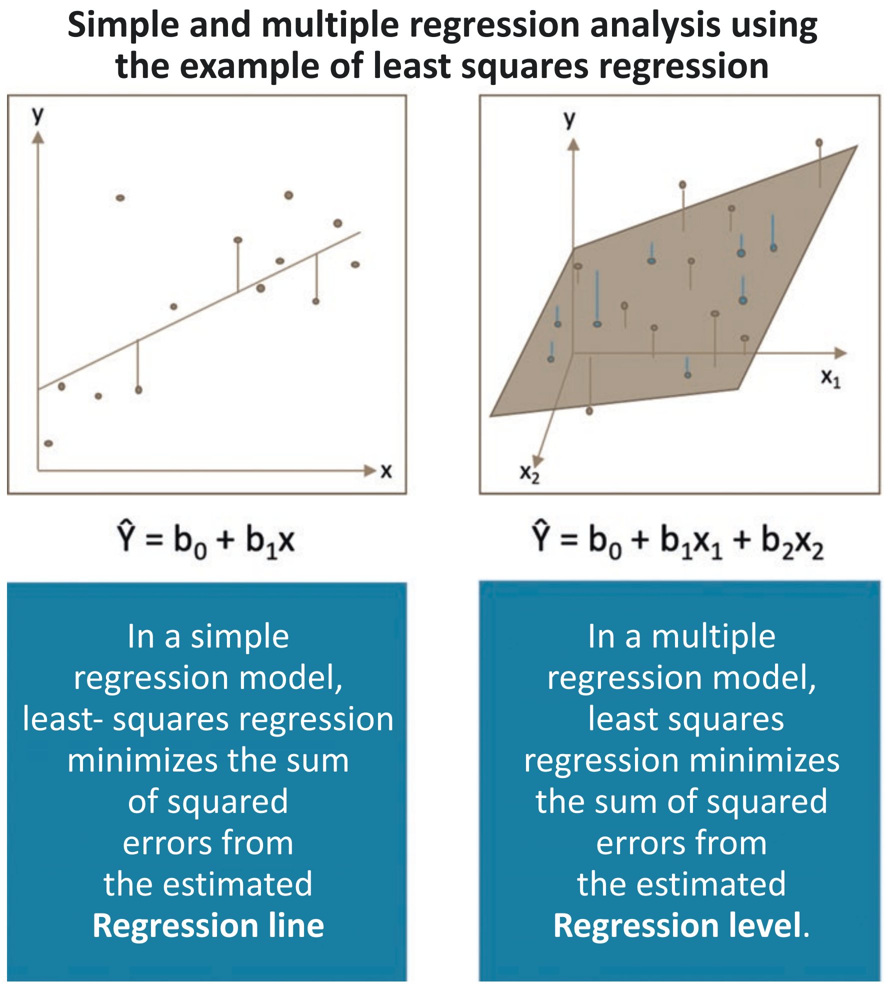
From the plot Fig. 2.13, it could be seen that there is a correlation between the life expectancy of women and the number of doctors in the population. This is probably true and one could say it is quite simple: provide more doctors in the population and life expectancy increases (left side of the figure). But the reality is that you would have to look at other factors, like the possibility that doctors in rural areas might have less training or experience. Or maybe they do not have access to medical facilities like trauma centers. Adding these additional factors would result in adding an additional dependent variable to the regression analysis, creating a model (right side of figure) for multiple regression analysis. The output differs depending on how many variables are present - but it is essentially the same type of output found in a simple linear regression.

Optimization

Optimization algorithms help to minimize or maximize an objective function E(x) (also called error function). This objective function represents a mathematical function that depends on the internal parameters of the model used in calculating the target values (Y) from the set of predictors (X) (Evans, 2017). Optimization algorithms generally fall into two main categories:

First-order optimization algorithms - These algorithms minimize or maximize an objective function E(x) based on its gradient values with respect to the parameters. The most widely used first-order optimization algorithm is gradient descent: the first-order derivative indicates whether the function is descending or ascending at a given point - basically, a line tangent to a point on its surface.

Stochastic gradient descent (SGD) is the simplest optimization algorithm to find parameters that minimize the given cost function. Apparently, the cost function should be convex so that the gradient descent is reduced to an optimal minimum. For demonstration



Simple and multiple regression analysis using the example of least squares regression

$\hat{Y} = b_0 + b_1x$

$\hat{Y} = b_0 + b_1x_1 + b_2x_2$

In a simple regression model, least- squares regression minimizes the sum of squared errors from the estimated **Regression line**

In a multiple regression model, least squares regression minimizes the sum of squared errors from the estimated **Regression level**.

Fig. 2.13 Regression and multiple regression analysis

purposes, imagine the graphical representation (see Fig. 2.14) of a hypothetical cost function.

It is started by defining some random initial values for the parameters. The goal of the optimization algorithm is now to find the parameter values that correspond to the minimum value of the cost function. Specifically, the gradient descent begins by computing derivatives for each of the parameters. These gradients then give a numerical fit to each parameter to minimize the cost function. This process continues until the local/global minimum is reached.

Second-order optimization algorithms - Second-order methods use the second-order derivative to minimize or maximize the objective function. The Hessian matrix serves as a

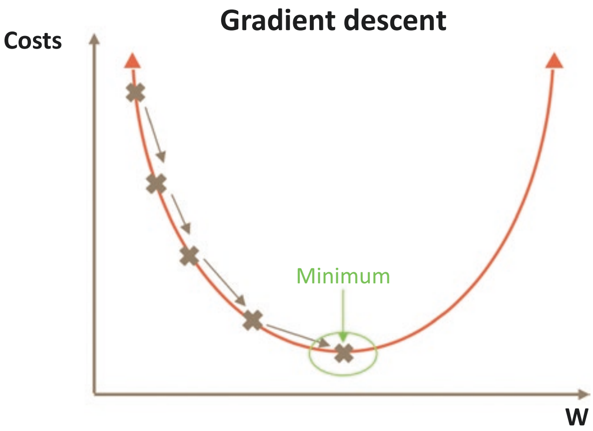


Fig. 2.14 Stochastic gradient descent (SGD) of a cost function

matrix of partial second-order derivatives.[10] Since the calculation of the second derivative is costly, the second-order is not often used. The second-order derivative tells whether the first derivative is decreasing or increasing - indicating the curvature of the function. Although the second derivative is costly to find and calculate, the advantage of a second-order optimization technique is that the curvature of the surface is not neglected or ignored.

Which Optimization Algorithm Should Be Used and When?

- First-order optimization algorithms are easier to compute and less time-consuming and converge fairly quickly on large data sets.
- Second-order techniques are faster only if the second-order derivative is known elsewhere. These methods are always slower and more expensive to compute, both in terms of time and memory.

Detection of Anomalies (Outliner)

Outlier analysis is a very important topic in predictive analytics and business analytics for two main reasons. First, outliers that need to be corrected or removed before starting to build a model can occur because of errors in the data. Second, outliers can occur naturally in the data because they are different from other values and therefore model building needs to account for these deviations, or it is these outliers that are the critical information points, or the information points that are sought after. Some algorithms are strongly affected by outliers. For example, the mean is affected by outliers, while the median is not. A simple

[10] For a comprehensive and scientifically sound presentation of second-order algorithms, see the University of Standford Lecture Notes by Prof. Ye. Available here: https://web.stanford.edu/class/msande311/lecture13.pdf.

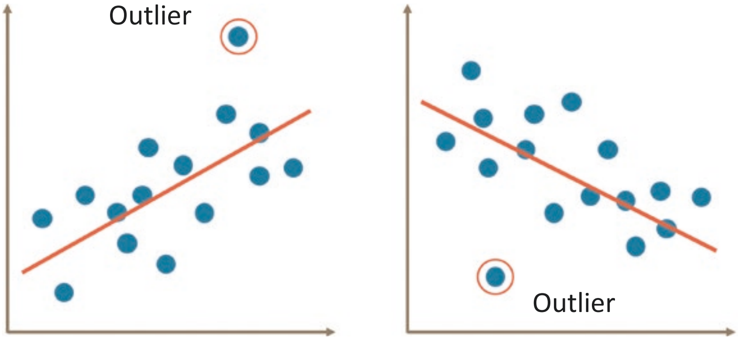


Fig. 2.15 Outliers in a regression

example is shown in Fig. 2.15, which depicts a linear regression where the effect of a single outlier would significantly change the line of best fit and model quality, and thus any prediction - both the value and confidence of the prediction.

With good data visualization, outliers can be detected to a certain extent but with very large and multidimensional data sets, this is limited due to the volume of data and the difficulty of visualizing these large data sets. Therefore, it is necessary that outlier detection algorithms automatically search and find unusual values.

The two main applications of outlier analysis reflect the two main reasons for such analysis. The first application is in the preliminary stage of building a model. Here, this analysis is used to identify errors in the data so that they can be corrected or ignored, or to identify unusual values that are valid and need to be accounted for in the model being built. Such outliers can affect the type of algorithm used in the analysis, as some algorithms are more sensitive to outliers than others. The second application area is the detection of outliers or anomalies, which is a key component in, for example, fraud analysis looking for unusual activity. This purpose is analogously applicable in a variety of areas: in production process control, fault detection, intrusion detection, fraud detection, system monitoring, and event detection. One of the biggest potentials definitely comes from applying the algorithms in fraud detection on the largest possible data sets (transaction data in the financial sector, for example). The volumes of data here are extremely large and can even be analyzed in real-time. The result of the outlier analysis is usually complemented by business rules that encapsulate the business logic. Example rules would be: “If the credit card usage is not in location A (country, region, etc.), then investigate this transaction”, or: “If an insurance claim is repeated x times, then this must be checked manually.”

Inter-quartile range test - The inter-quartile range test (IQR), also known as the Tukey test [21], named after its author John Tukey, is a simple yet robust test for identifying numerical outliers. It is the computational basis behind the construction of the so-called “box plots” (see Fig. 2.16) produced by the test to identify outliers. In simple mathematical terms, the formula for the inter-quartile range test is IQR = Q3 - Q1.

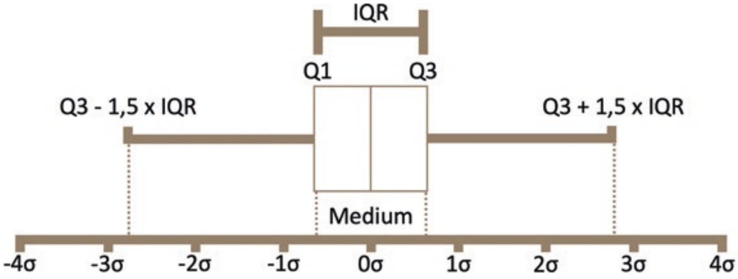


Fig. 2.16 Schematic representation of the inter-quartile range test

The IQR can also be taken as a measure of the distribution of values because statistics first assumes that values are grouped around a central value. The IQR indicates how distributed the "mean" values are. The value can also be used to see if some of the values are "too far" from the central value. These points that are too far away are called "outliers" because they are "outside" the range where these would normally be expected. The IQR is the length of the box in the box-and-whisker plot (box plot or box graph) [21]. An outlier is any value that is more than one and a half times the length of the box from either end of the box. That is if a data point is below Q1–1.5 × IQR or above Q3 + 1.5 × IQR, it is considered too far from the central values to be appropriate.

Questions

> Why is the width for the outliers one and a half times that of the box? Why does this value show the difference between "acceptable" and "unacceptable" values? When John Tukey invented the box-and-whisker plot [21] to show these values in 1977, he chose 1.5 × IQR as the demarcation line for outliers. This worked well, so this value has been used ever since. If you look deeper into statistics, you will find that this measure of reasonableness for bell-shaped data means that usually, only about one percent of the data will ever be outliers.

Recommendation or Recommender Systems

A recommender system is a system that aims to predict a user's evaluation of a particular artifact (product, movie, message, etc.) based on methods or algorithms [22]. Recommender systems have found applications in various fields. For example, companies such as Amazon, Netflix, Linkedin, or Spotify use recommender systems to help users discover new and relevant artifacts (products, videos, jobs, or music), thus serving to generate a positive user experience (directly) and thus increase sales (indirectly).

In the following, the three main approaches are presented: the collaborative ("Collaborative filtering"), the content-based ("Content-based filtering"), and the hybrid approach.

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} \end{bmatrix} \begin{bmatrix} p_{11} & ? & p_{13} & ? & p_{15} \\ ? & ? & ? & p_{24} & ? \\ p_{31} & ? & p_{33} & ? & ? \\ ? & p_{42} & ? & p_{44} & p_{45} \end{bmatrix}$$

Fig. 2.17 Schematic representation of a user-item matrix

Fig. 2.18 Cosine similarity

$$similarity(a,b) = cos(a,b) = \frac{a \cdot b}{||a|| * ||b||}$$

Collaborative filtering - Collaborative filtering systems make recommendations based on known users' preference for items (clicked, viewed, purchased, liked, rated, etc.). The preference can be represented as a user-item matrix (see Fig. 2.17). Below is an example of such a matrix describing the preference of 4 users on 5 items. Where p{12} is the preference of user 1 for item 2 (left side of the matrix). Although the entries can be numeric, e.g., for a movie rating prediction on Amazon Prime Video, the rating scale from 1 to 5 xwould be included but in most applications, these entries are purely binary (e.g., clicked, watched, bought). In reality, this user-item matrix is several million $*$ million dimensions large (e.g., Amazon, Youtube) - and most entries are missing (right side of the matrix)!

The goal of recommender systems is to fill these missing entries. Collaborative filtering is based on the assumption that people who have shown a preference in the past will repeat that preference in the future, and this represents the probability of similar items [23]. There are several collaborative filtering approaches but the most intuitive is nearest-neighbor classification ("nearest-neighbor"):

Nearest-neighbor classification - These methods are based on the similarity between pairs of elements (here: users or customers). The cosine similarity used here (see Fig. 2.18) is often used to measure the distance between two vectors.

The preference matrix (see Fig. 2.19) can be represented in terms of item vectors. The similarity between item I1 and item I2 is calculated as cos(I1,I2). The matrix can also be represented as user vectors (right side of the figure).

The similarity between U1 and U2 is calculated as cos(U1,U2). It is important to note that the missing values in the preference matrix are often filled with zeros.

For user i, the items that are preferred by the most similar users of user i (user-to-user) or the most similar items of the artifacts associated with user i (item-to-item) can be recommended.

Fig. 2.19 Preference matrix and user vectors

$$P = [I_1 \;\; I_2 \;\; I_3 \;\; I_4 \;\; I_5] \qquad P = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_1 \end{bmatrix}$$

Item-to-item approaches are widely used in practice, for example by Amazon,[11] Youtube,[12] Linkedin[13], and many others. When a customer “likes” an item, an item-to-item system can quickly find similar items (similar items for each item are pre-computed and stored in a key-value data store). In addition, item-to-item recommendations can be more interpretable than user-to-user recommendations, e.g., systems can explain why an item is recommended (“because the user also likes product Y”). It is possible that the number of items that are similar to an item is too small (for example, if a threshold is used on the similarity scores). Therefore, one approach is to expand the list of similar items by including the similar items of those items as well.

Content-based filtering - Content-based filtering is the second commonly used approach in the development of recommender systems. A content-based recommender system uses keywords to describe the different artifacts and the profile of the users is designed to explicitly specify preferences. Thus, the algorithms suggest items that are identical to those that have been preferred by users in the past [27].

The underlying idea here is this: if a user indicates that he “likes” an artifact (by clicking on it, rating it positively, or viewing it more often), this means that there is a high probability that he will want to use it (buy, view) the artifact. The approach for recommending other items to this user is to select items with similar attributes or descriptive properties (brand, color, features, ...).

The steps for recommending products or content to a user in content-based filtering are as follows:

- Identify factors (descriptors) that describe and distinguish the artifacts.
- Collect all other elements that are similar in relation to these descriptors.
- For each product, create a tuple or number vector that represents the strength of each descriptor for the product.
- Creating a user profile based on history (purchase history, click history, etc.). It has the same number of factors and their strength would indicate how much influence the user has on that factor. These weights indicate the importance of each feature to the user and can be measured from separately scored content vectors based on different methods.
- Recommend elements that are closest in terms of these factors. The simple methods use averages associated with the evaluated item’s vector, while complicated methods use machine learning, such as artificial neural networks, decision nodes, cluster analysis, and Bayes classifiers to estimate the probability that a user prefers an item.

[11] See: [24].

[12] See: [25].

[13] See: [26].

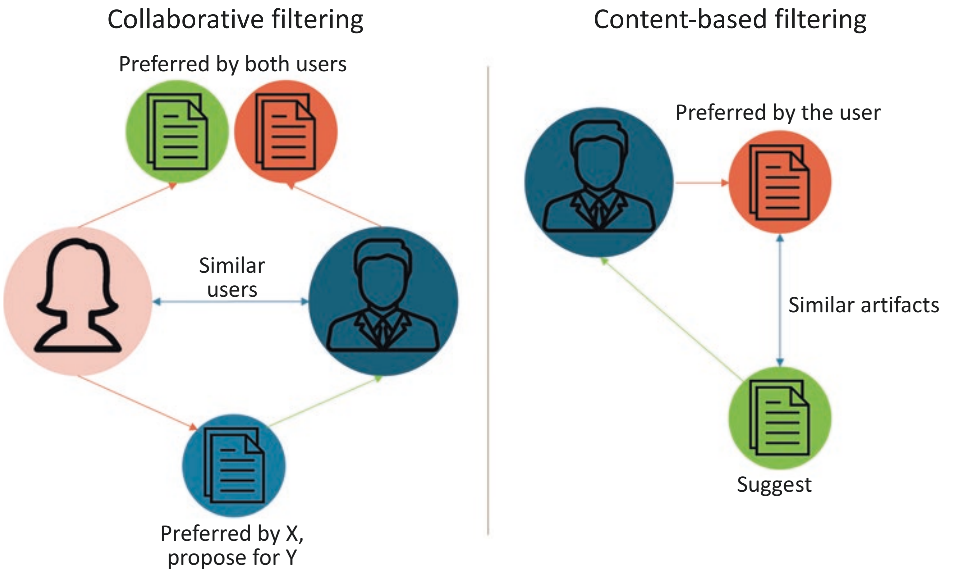


Fig. 2.20 Schematic representation of collaborative and content-based filtering

Hybrid Recommender Systems

Recent studies [25, 28] show that using a hybrid approach, i.e., combining content-based and collaborative filtering (see also the comparative illustration in Fig. 2.20), can be even more effective. These methods can be used in different ways - by first applying collaborative and content-based predictions separately and combining them later, or by combining all approaches into a single model. These combined solutions also help with the more common problem in recommender systems, such as a lack of data.

Netflix is the best example of a hybrid recommendation system. The website's recommendations are based on the general search and watching habits of similar users [26, 29] and then augmented and improved with the individual's behavior.

When to Use Which Algorithm?

It is a very difficult task for less experienced users to decide which algorithm to use in an analysis. Considering that the programming language "R", which is widely used by data scientists in research and practice, has over 3.500 different packages or algorithms to choose from, the choice can indeed be overwhelming. This chapter addresses the question of which algorithm to use and when. It begins by describing the most important factors that should be considered when choosing an algorithm.

Two main factors determine this decision:

Table 2.2 Overview of tasks and algorithms

Task	Algorithm category	Example algorithms
Detection of outliers and anomalies	Statistics	Variance test, IQR test, anomaly detection
Data preparation before analysis	Data preparation	Sampling, scaling, binning
Statistical inference	Sampling theory	T-tests, F-tests, ANOVA
Relationships, cause, and effect	Correlation and regression	Multiple linear regression, nonlinear regression
Grouping of data (clustering)	Cluster analysis	ABC analysis, K-means, Kohonen SOM
Forecast/time series forecast	Time series analysis	Exponential smoothing, regression
Association or affinity analysis	Association analysis	Apriori
Prediction, modeling	Classification analysis	Decision trees, neural networks, regression models
Social network analysis	Network analysis	Jaccard's coefficient, common neighbors
Optimization	Optimization	Linear and non-linear programming
Risk analysis, modeling	Simulation	Monte Carlo analysis

Assignment of tasks to algorithm categories and examples

1. What is the goal? For example, grouping data, searching for links in the data, or forecasting a time series of data values.
2. What data is available and what attributes does this data have? For example, are they numeric, categorical, or boolean?

From this, an algorithm or a selection of algorithms can then be derived and the question that follows is: which algorithm fits best? This is the key question in itself, as there can be multiple evaluation criteria and in some cases, there is no right answer at all. For example, in cluster analysis, since there is no best clustering in the sense of optimizing the model parameters to predict a target variable. In addition to the question of which algorithm provides the best fit, there is the important consideration of how useful the derived model is in practice. Complexity does not necessarily equate to usefulness. With respect to the first question, Table 2.2 can be referenced with the columns of the task, algorithm category, and example algorithms.

For example, in the table, one can see that in the task of searching for unusual values or outliers identified as outlier detection, algorithms such as the variance test or the interquartile range test can be used. If the task is to build a predictive model where one variable is to be predicted from the data of other variables (the prediction, model-building task), as in churn analysis or target marketing, then examples of groups of relevant algorithms include decision trees, neural networks, and regression models. Within these three groups, there are many different algorithms, and the decision of which algorithm to use depends on many data-specific reasons. Therefore, it may be advisable to test with several algorithms and then make a detailed decision based on the results (goodness or

Table 2.3 Overview of the classes of applications, variables, and algorithms

Problem class	Input (independent variable)	Output (dependent variable)	Algorithm
Dependence/ association	Categorical	Categorical	Apriori, Apriori Lite
Clustering	Numeric	–	k-means, ABC analysis, Kohonen SOMs
Classification (regression)	Numeric	Numeric	Multiple and non-linear regression
Classification (regression)	Numeric/Categorical	Numeric/ Categorical	Logistic regression
Classification (decision tree)	Numeric/Categorical	Numeric/ Categorical	C4.5, CHAID
Classification (neural networks)	Numeric/Categorical	Numeric/ Categorical	Neural networks
Classification (other)	Numeric	Numeric/ Categorical	K-nearest-neighbor
Time series	Numeric	Numeric	Regression, exponential smoothing
Outlier detection	Numeric	–	IQR, variance test

A generic overview of the assignment of problem classes to algorithms based on the dependent and independent variables

correctness of the results). One can also make additional considerations, such as how easy it is to use the results of the analysis in a business process.

Knowledge of the individual algorithms is clearly an advantage but not mandatory to benefit from predictive analysis. As mentioned, today's frameworks and services (section "SQL") make it very easy to try out and test more than one algorithm from the set of relevant algorithms that yield the best results. Moreover, in many cases, the algorithm does not need to be understood in every detail in order to use it in a project. This is of course a debatable point and any knowledge is clearly beneficial. Also, the kind of data (numeric, categorical) one has or expects as input and output is a crucial aspect (Table 2.3).

One possible analogy to describe this is although not many people know how airplanes manage to fly, people still use them regularly. Although the analogy is not perfect, it is intended to show that we have a certain confidence in flying simply from the observation that the process has a high success rate. In this discussion, it is important to remember that a major challenge in the analysis is first identifying, obtaining, reviewing, and preparing the data for analysis. Finding the best model is certainly important but it is not the biggest challenge. The second major challenge is the transition from analysis to implementation and integration of the analyses into business processes (see the process model introduced above). So, to decide which algorithm should be used and when one can simply apply all the appropriate algorithms to the data and choose the best one. This is a logical and reasonable approach but it raises the question: what is "best" and how is it measured? The answer to this question varies by algorithm group. For example, for association analysis, the choice of algorithms falls on Apriori and Apriori Lite - the latter being a subset as it is limited to finding individual pre- and post-rules. The choice, therefore, has more to do with rule requirements and performance, as Apriori Lite will be faster than generic Apriori but again is limited in terms of rules extracted from the data. For cluster analysis, the concept of how best to select what is harder to define. For ABC analysis, you cannot really say that different values of A, B, or C are better or worse. It is up to the user to choose what is best for them - there is no concept of optimal model fit. The k-means algorithm does not necessarily provide better cluster analysis than "Kohonen self-organizing maps" (Kohonen SOMs) and vice versa. However, k-means is easier to understand, hence its popularity. Although Kohonen SOMs are complex, they are more flexible in use because there is a less enforced assignment of records to a cluster (in the sense that k-means must have k-clusters, while Kohonen SOMs do not specify the number of clusters). There are quality metrics for clusters but these metrics are more indicators than definitive computations. The best approach to evaluating algorithms is to try both k-means and Kohonen SOMs with different numbers of clusters to examine the solutions and decide which is best for the application.

The concept of the best algorithm also varies within each subset of classification analysis. However, there are basically two types of predictions from the classification model: numerical or categorical. For numerical predictions, the most common measure is the mean squared error for each data point, also represented as the mean squared error (MSE). In estimation theory, this value indicates how much a point scatters around the value being estimated, and thus the MSE is a key quality criterion for estimators. In regression analysis, the MSE is interpreted as the expected squared distance an estimator has from the true value. There is the associated quadratic mean (root mean square or RMS). This is the mean calculated as the square root of the quotient of the sum of the squares of the observed numbers and their number. The RMS is used by regression analysis, which provides numerical predictions, including statistical measures of goodness of fit, such as R-squared, analysis of variance (ANOVA), and the F-value. Categorical predictions are generally evaluated using what is called confusion matrices, which are essentially designed to show how many times each category was predicted correctly and how many times incorrectly. Based on the matrix, there are then model quality measures, such as sensitivity or true positive rate and specificity or true negative rate. For binary classification models, we can plot and compare model performance in gain and lift diagrams. For time series analysis, the same quality measures apply as for numerical predictions in classification analysis, except that the analysis is overtime periods. For the outlier tests, the variance test and the inter-quartile range (IQR) test are used to look for overall outliers in the data set. The variance test is trivial but it is affected by the outliers themselves. Therefore, the more popular IQR test is used because it takes into account the median and quartile as a measure to identify an outlier and thus is not influenced by the actual outliers themselves. This anomaly detection algorithm is used to find local outliers in the data set.

From the above can be derived as rules to help in the selection of algorithms:

- When looking for associations in the data and
  - if multiple assignments of elements are desired, then one uses Apriori,
  - only single rules are desired, then use Apriori Lite,
  - if the performance of Apriori is too slow, you should switch to Apriori Lite Sampling.
- When searching for clusters or segments in the data and when
  - the cluster sizes are user-defined, then ABC analysis can be used,
  - the desired number of clusters is known, then k-means is used,
  - the number of clusters is unknown, one uses Kohonen SOMs.
- If the data is to be classified and the target variable is numeric, there is only one independent numeric variable, and if
  - a relationship is to be taken into account, bivariate linear regression is used,
- otherwise, if
  - a non-linear relationship is to be considered, one uses a bivariate exponential or geometric or natural logarithmic regression, there is more than one independent numerical variable, one uses multiple linear and non-linear regressions for linear and non-linear models.
- If one is looking for classification data and the variables are categorical or a mixture of categorical and numeric and if
  - the output of decision tree rules is desired, then one uses either C4.5, CHAID, or CNR and decides according to the best result,
  - the output of the probability of a result is preferred, then one uses logistic regression,
  - the model quality is in the foreground and the understanding of the model is less important, then one uses neural networks and decides according to the best result.
- If time-series data is to be predicted and if the data is to be
  - are constant or stationary, then one uses single exponential smoothing,
  - represent a trend, then double exponential smoothing is used,
  - are seasonal, then one uses triple exponential smoothing.

References

1. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach Prentice Hall Series in Artificial Intelligence, vol. xxviii, p. 932. Prentice Hall, Englewood Cliffs (1995)
2. Watson, H.J., Rainer Jr., R.K., Koh, C.E.: Executive information systems: a framework for development and a survey of current practices. MIS Q. **15**, 13–30 (1991)
3. Goodfellow, I., et al.: Deep Learning, vol. 1. MIT Press, Cambridge (2016)
4. Amirian, P., Lang, T., van Loggerenberg, F.: Big Data in Healthcare: Extracting Knowledge from Point-of-Care Machines. Springer, Cham (2017)
5. Zachman, J.A.: A framework for information systems architecture. IBM Syst. J. **26**(3), 276–292 (1987)
6. Sowa, J.F., Zachman, J.A.: Extending and formalizing the framework for information systems architecture. IBM Syst. J. **31**(3), 590–616 (1992)

7. Witten, I.H., et al.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Cambridge (2016)
8. Gorry, G.A., Scott Morton, M.S.: A framework for management information systems. Sloan Manag. Rev. **13**, 55–70 (1971)
9. Sprague Jr., R.H.: A framework for the development of decision support systems. MIS Q. **4**, 1–26 (1980)
10. Robert, C., Moy, C., Wang, C.-X.: Reinforcement learning approaches and evaluation criteria for opportunistic spectrum access. In: 2014 IEEE International Conference on Communications (ICC), IEEE (2014)
11. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)
12. Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach. Springer Science & Business Media, Berlin (2013)
13. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB (1994)
14. Gluchowski, P., Chamoni, P.: Analytische Informationssysteme: Business Intelligence-Technologien und -Anwendungen, 5th edn. Springer Imprint/Springer Gabler, Berlin/Heidelberg (2016)
15. Bollinger, T.: Assoziationsregeln – Analyse eines Data Mining Verfahrens. Informatik-Spektrum. **19**(5), 257–261 (1996)
16. Decker, R.: Empirischer Vergleich alternativer Ansätze zur Verbundanalyse im Marketing. Proceedingsband zur KSFE. **5**, 99–110 (2001)
17. Dhanachandra, N., Manglem, K., Chanu, Y.J.: Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Comput. Sci. **54**, 764–771 (2015)
18. Kim, N., et al.: Load profile extraction by mean-shift clustering with sample Pearson correlation coefficient distance. Energies. **11**, 2397 (2018)
19. Larcheveque, J.-M.H.D., et al.: Semantic clustering. Google Patents (2016)
20. Chatterjee, S., Hadi, A.S.: Regression Analysis by Example. Wiley, New York (2015)
21. Tukey, J.W.: Comparing individual means in the analysis of variance. Biometrics. **5**(2), 99–114 (1949)
22. Yu, C.H.: Exploratory data analysis. Methods. **2**, 131–160 (1977)
23. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The Adaptive Web, pp. 325–341. Springer, Heidelberg (2007)
24. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**, 76–80 (2003)
25. Gunawardana, A., Meek, C.: A unified approach to building hybrid recommender systems. RecSys. **9**, 117–124 (2009)
26. Liu, N.N., Zhao, M., Yang, Q.: Probabilistic latent preference analysis for collaborative filtering. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM (2009)
27. Gong, S.: A collaborative filtering recommendation algorithm based on user clustering and item clustering. JSW. **5**(7), 745–752 (2010)
28. Burke, R.: Hybrid recommender systems: survey and experiments. User Model. User-Adap. Inter. **12**(4), 331–370 (2002)
29. Zhao, X., Zhang, W., Wang, J.: Interactive collaborative filtering. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, ACM (2013)

AI and BA Platforms

3

Basic Concepts and Software Frameworks

Data Management

When the term data management is mentioned, one immediately thinks of the classic database and the relational database model. But here, too, technical progress has made the choice considerably more complicated in recent years. In the context of BA projects, however, it is first necessary to distinguish between two basic approaches to data management: the data warehouse (comes closest to the classic database) and the data lake.

Data Warehouse

A data warehouse (DWH) resides in a database, and it does not matter which database forms the basis from Redshift and BigQuery to MySQL and Postgres. A data warehouse is a central repository for data collected from one or more data sources. With the help of data warehouses, it is possible to manage data and perform quick analyses on large data sets to uncover hidden patterns. DWHs store current and historical data and are used to create analytics reports for data consumers across the enterprise. Examples of reports can range from annual financial reports to hourly revenue analysis trends.

Schema-On-Write

Schema-on-write has been the standard for the process of data storage in relational databases and thus also for data warehouses for many years. Before data is written to the database, the structure of this data is strictly defined and metadata is stored and updated. Irrelevant data is discarded, data types, lengths, and positions are described in advance. The schema, i.e. the columns, rows, tables, and relationships, is defined in advance and during construction for the specific purpose for which the database will serve. Then the

F. Weber, *Artificial Intelligence for Business Analytics*,
https://doi.org/10.1007/978-3-658-37599-7_3

data is put into the predefined structure and stored. Thus, the data must all first be cleansed, transformed, and adapted to this structure before it can be stored in a process. This is commonly referred to as ETL (extract transform load). This is why the paradigm here is also called “schema-on-write“since the data structure is already defined when the data is written and stored.

When Should a DWH Be Used?

- If the data sources are relatively constant.
- When data queries are known in advance for the most part.
- When data can be modeled into a schema structure.
- When a high degree of data accuracy is required, e.g. for accounting purposes.
- When strict access control and a high level of security are required.

Which Platforms Can Be Used to Implement a DWH?

- Relational database management systems (RDBMS): MySQL, Oracle, SQL Server, Postgres, SAP HANA, etc.
- Columnar databases: Vertica, ParAccel, Amazon Redshift, Google BigQuery, SAP HANA

Data Lake

Data lake is a relatively new term attributed to James Dixon, the CTO of Pentaho [1]. Essentially, a data lake is a large storage location for raw or lightly processed data in its original format. The data lake stores data in a flat structure, usually as files. Data in the data lake is associated with a unique ID and tagged with metadata. When a query arises, the data lake can be queried for relevant data and this smaller data set can then be analyzed to answer the question. Hadoop, Google Cloud Storage, Azure Storage, and the Amazon S3 platform can be used to build data lake repositories. Data lakes do not require much planning - typically there is no schema and no ETL process. Thanks to the falling cost of data storage both on-premises in your own data centers and the cloud and the abundance of virtual services, a data lake can be set up quickly. Even before anyone knows what questions they want to ask in the future, data from a variety of sources and in a variety of formats can be stored immediately in the data lake. However, because data lakes contain a variety of data formats and large amounts of data, querying is much more difficult. Traditional BI tools usually do not support a Data Lake at all - this then requires transformation to generate insights. This makes the enterprise data lake a playground for people with advanced data skills (data scientists and experienced developers) but less accessible to business users.

The paradigm for storing data in a data lake is called **schema-on-read.** This describes the concept that you do not need to know in advance what you will do with data in the future. Data of all types, sizes, shapes and structures can all be “unthinkingly thrown” into the data lake and other Hadoop data storage systems. While some metadata, data about that data, needs to be stored so that you still know what is in it at the end, you do not need to know how the content will be structured. It is quite possible that the data stored for one purpose will even be used for a completely different purpose than originally intended. The data is stored without first deciding what information is important, what should be used as a unique identifier, or what part of the data needs to be summed and aggregated to be useful. Therefore, the data is stored in its original granular form, with nothing thrown away because it is now considered unimportant, nothing is aggregated into a composite, and there is no key information or dependencies (indexes). In fact, no structural information is defined at all when the data is stored. When someone is ready to use that data, they define at that time which parts are essential for their purpose. All that is defined in advance is where to find the information that is important for that purpose and what parts of the data set can be ignored. This is why this paradigm is also called “schema-on-read“because the schema is defined at the time of reading and using the data, not at the time of writing and storing it.

When Should a Data Lake Be Used?

- When future user-cases and requirements for analytics are unknown, but the company identifies data analytics as an important competitive differentiator.
- When data must be collected immediately and there is no time for planning.
- When data sources and formats are highly dynamic.
- If the data is too large to be stored in a database for budgetary reasons.
- When analytical queries are not known in advance or change frequently.
- When data scientists need a “playground” to find and develop new insights.
- When more groups of people in a company need access to the data.

Which Platforms Can Be Used to Implement a Data Lake?

- Hadoop Distributed File System (HDFS)
- Amazon Simple Storage Service (S3), Google Cloud Storage, Azure Data Lake Storage
- Data lakes can also be combined with a classic data warehouse, HBase, or a NoSQL database (such as MongoDB).

Data lakes and data warehouses are both used for storing large amounts of data but each approach has its applications. Typically, a data warehouse is a relational database housed on on-premise servers or the cloud, although this has been less common due to the amount of data involved. The data stored in a data warehouse is extracted from various online transaction processing (OLTP) applications to support business analytics queries and data marts for specific internal business groups, such as sales or marketing organizational units.

Data warehouses are useful when there is a large amount of data from operational systems that need to be readily available for analysis. Because the data in a data lake is often uncurated and can come from sources outside the company's operational systems, data lakes are not well suited for the average business analytics user. Nevertheless, they have the advantage that even without a predefined reason and process, data is kept for use cases that will be created in the future. Table 3.1 shows an overview that can be used as a basis for decision-making.

Data Streaming and Message Queuing

Stream processing is the processing of data "in motion" (stream), i.e., the computation on and analysis of data directly as it is generated or received (see also the comparison to traditional data processing in Fig. 3.1). This paradigm takes into account the fact that most data originates as continuous streams: sensor events, user activity on a website, or financial trading data. All of this data is generated as a series of events over time. Big data, in the sense of big data, has determined the value of insights from processing data. Such insights are not all equal. Some insights are more valuable shortly after it happens, as the value decreases very quickly over time. Stream Processing enables such scenarios and delivers insights faster, often within milliseconds to seconds of the trigger.

Thereby, many names exist real-time analysis, streaming analysis, complex event processing (CEP), real-time streaming analysis, and event processing. Although some terms have historically had differences, most tools (frameworks) now appear under the umbrella term stream processing. Prior to stream processing, this data was often stored in a database, file system, or other forms of bulk storage. Applications would query the data or compute over the data as needed. Stream processing itself flips this paradigm: the application logic, analytics, and queries are continuous, and the data flows through them continuously. Upon receiving an event from the stream, a stream processing application responds to that event: it may trigger an action, update an aggregate or other statistic, or "remember" that event as a future reference. Streaming computations can also process multiple streams together, and each computation over the event stream can generate other event streams.

Table 3.1 Comparison between data warehouse and data lake

	Data warehouse	Data lake
Data	Structured and processed	Structured to unstructured (raw data)
Processing	Schema-on-write	Schema-on-read
Storage	Hierarchically structured	Object based, no structure
Flexibility	Rather rigid	Highly flexible
Security	Mature	Not yet fully developed
Target group	Business customers, business users, specialist departments	Data scientists

A data warehouse is very different from a data lake

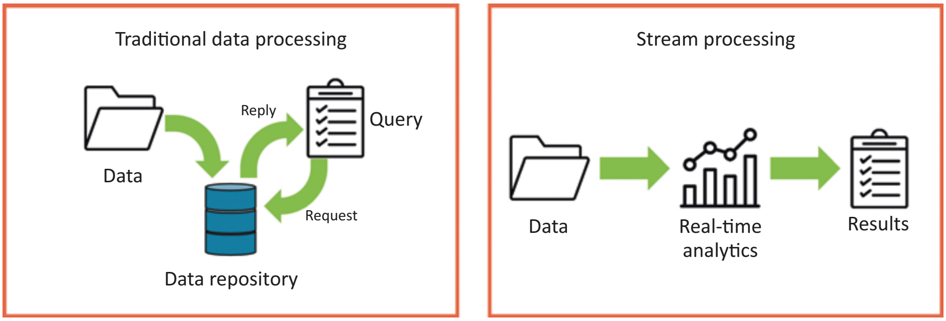


Fig. 3.1 Comparison between traditional data processing and stream processing

Table 3.2 Comparison between batch and streaming

	Batch	Streaming
Scope of data	Query or process over all or most of the data in the dataset	Querying or processing data within a rolling time window or only on the last data record
Data sets	Large amounts of data	Individual records or micro-batches consisting of a few records
Performance	Latencies in minutes to hours	Requires latency on the order of seconds or milliseconds
Analyses	Complex analyses	Simple response functions, aggregates, and rolling key figures

An evaluation framework for deciding between batch and streaming can be built on the levels of data size, data volume, performance, and analysis

The systems that receive and send the data streams and execute the application or analysis logic are called stream processors. The main task of a stream processor is to ensure that the data flow and computation scales are efficient and fault-tolerant.

Before looking at streaming data, it is worth comparing stream processing and batch processing (see Table 3.2). Batch processing can be used to compute arbitrary queries over different data sets. It typically computes results derived from all the data it encompasses and allows for in-depth analysis of large datasets. MapReduce-based systems, such as Amazon EMR are examples of platforms that support batch jobs. In contrast, stream processing requires taking a sequence of data and incrementally updating metrics, reports, and summary statistics in response to each incoming data set. And, thus, stream processing is better suited for real-time monitoring and response functions.

Many companies are now building on hybrid models by combining the two approaches and creating a real-time (streaming) layer and a batch layer simultaneously or in series. Data is first processed by a streaming data platform to provide real-time insights and is then loaded into a data store where it can be transformed and used for a variety of batch processing use cases.

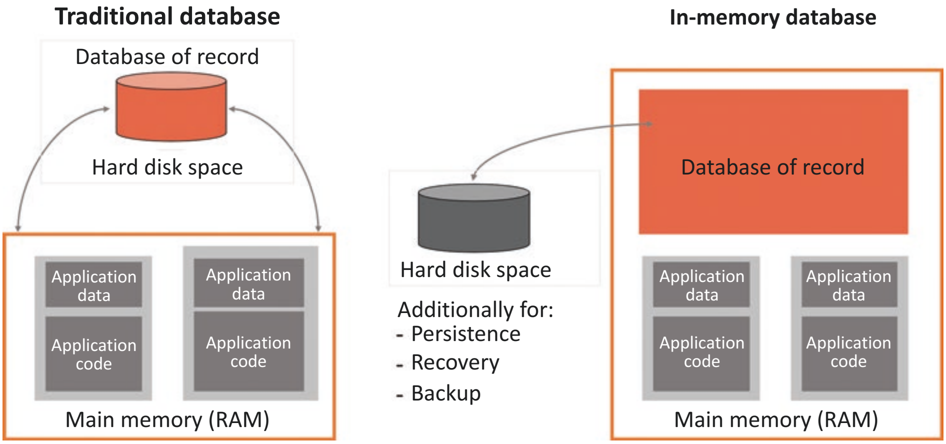


Fig. 3.2 Comparison between the traditional database and the in-memory database

Database Management System

An in-memory database management system (IMDBMS) is a database management system (DBMS) that relies predominantly on main memory for data storage, management, and manipulation. The comparison to traditional databases is shown schematically in Fig. 3.2. This eliminates the latency and overhead of disk storage and reduces the instruction set required to access data. For more efficient storage and access, data can be stored in a compressed format.

Traditional DBMSs move data from disk to main memory into a cache or buffer pool when it is accessed. Moving data to the main memory makes re-accessing the data more efficient but the constant need to move data can cause performance problems. Since the data in an IMDBMS is already in memory and does not need to be moved, application and query performance can be greatly improved. To ensure the persistence of data in an IMDBMS, it must be periodically moved from memory to persistent, nonvolatile storage. This is important because data stored in memory would not survive a failure (main memory cannot store data in the event of a power failure). There are several ways to achieve this data persistence. One way is transaction logging, in which periodic snapshots of the in-memory database are written to nonvolatile storage media (hard disks). If the system fails and needs to be restarted, the database can then be rolled back or redirected to the last completed transaction. Another way to maintain data persistence is to create additional copies of the database on nonvolatile media. At the hardware level, there is also the option of using non-volatile RAM (NVRAM), such as battery RAM backed up by a battery, or ferroelectric RAM (FeRAM), which can store data when powered off. Hybrid IMDBMSs that store data on both hard disks and memory chips are also conceivable and available.

In-memory database systems have a wide application but are mainly used for real-time applications that require high performance. Use cases for IMDBMS are applications with real-time data management requirements, such as telecommunications, finance, defense, for decision support and optimization. Applications that require real-time data access, including call center applications, travel and reservation applications, and streaming applications, are also good candidates for use with an IMDBMS.

Apache Hadoop

Hadoop describes itself as an “open-source distributed processing framework” that enables data processing and storage for big data applications in clustered systems. It is at the center of a growing ecosystem of big data technologies used primarily to support advanced analytics - particularly predictive analytics, data mining, and machine learning applications. Hadoop can handle various forms of structured and unstructured data and provides users with more flexibility to collect, process, and analyze data than relational databases and data warehouses.

Hadoop‘s primary focus here is on analytic applications, and its ability to process and store different types of data makes it a particularly good choice for big data analytics applications. Big data environments typically include not only big data but also various types of structured transactional data to semi-structured and unstructured forms of information, such as clickstream records, web server, and mobile application logs, social media posts, customer emails, and sensor data from the Internet of Things (IoT). Originally known only as Apache Hadoop, the technology is being developed as part of an open-source project within the Apache Software Foundation (ASF). The commercial distribution of Hadoop is currently offered by four primary big data platform providers: Amazon Web Services (AWS), Cloudera, Hortonworks, and MapR Technologies. In addition, Google, Microsoft, and other vendors offer cloud-based managed services based on Hadoop and related technologies.

Hadoop runs on clusters of standard servers, and it supports thousands of hardware nodes for massive amounts of data. Hadoop uses a namesake distributed file system that enables fast data access across nodes in a cluster, as well as fault-tolerant features to allow applications to continue running when individual nodes fail. Consequently, Hadoop became a foundational data management platform for big data analytics applications after its emergence in the mid-2000s.

Hadoop was developed by computer scientists Doug Cutting and Mike Cafarella [2], initially to support processing for a Dutch open-source search engine and associated web crawler. After Google published technical papers in 2003 and 2004 detailing the Google File System (GFS) and MapReduce programming framework, Cutting and Cafarella modified earlier technology plans and developed a Java-based MapReduce implementation and file system modeled on Google. In early 2006, these elements were spun off from Nutch and became a separate Apache subproject that Cutting named Hadoop after his son’s stuffed elephant. At the same time, Cutting was contracted by internet service provider Yahoo, which became Hadoop‘s first production user later in 2006.

The core components in the first iteration of Hadoop were MapReduce, the Hadoop Distributed File System (HDFS), and Hadoop Common, as well as a set of common tools and libraries. As its name implies, MapReduce uses the “map and reduce” paradigm to split processing jobs into multiple tasks that execute on the cluster nodes where the data is stored, and then combine what the tasks produce into a cohesive set of results. MapReduce initially acted both as Hadoop‘s processing engine and as a cluster resource manager, connecting HDFS directly to the system and restricting users to running MapReduce batch applications (Table 3.3).

This changed with the release of Hadoop 2.0, which became generally available in 2013 in version 2.2.0. It introduced Apache Hadoop YARN, a new cluster resource management, and job scheduling technology that inherited these features from MapReduce. YARN - short for Yet Another Resource Negotiator but typically just referred to by the acronym - ended the strict reliance on MapReduce and opened up Hadoop to other processing engines and various applications besides batch jobs.

Simply put, Hadoop has two main components. The first component, Hadoop Distributed File System, helps to share the data, put it on different nodes, replicate and manage it. The second component, MapReduce, processes the data from each node in parallel and computes the results of the job.

With YARN, the capabilities that a Hadoop cluster can deliver have greatly expanded - to include stream processing and real-time analytics applications that can run alongside processing engines like Apache Spark and Apache Flink. For example, some manufacturers are using real-time data that feeds into predictive maintenance applications in Hadoop to detect equipment failures before they occur (predictive maintenance). With fraud detection, website personalization, and customer satisfaction scoring, other real-time use cases are known and can be implemented on Hadoop.

Because Hadoop can process and store such a wide range of data, it allows organizations to set up data lakes as sprawling repositories for inbound information streams. A Hadoop data lake often stores raw data in such a way that data scientists and other analysts can access the full data sets as needed. Data lakes generally serve different purposes than traditional data warehouses, which contain cleansed transactional data. In some cases, however, companies consider their Hadoop data lakes to be modern data warehouses. Either way, the growing role of big data analytics in business decisions has made effective data governance and data security processes a priority when deploying data lakes.

Table 3.3 Overview of Hadoop and adjacent technologies

Hadoop File System (HDFS)	YARN (Yet another resource negotiator)	MapReduce	Hadoop Common
A file system that manages storage and access to data distributed across the carious nodes of a Hadoop cluster	Hadoop‘s cluster resource manager, responsible for allocating system resources to applications and scheduling jobs	A programming framework and processing engine used to run large batch applications in Hadoop systems	A set of utilities and libraries that provide the basic functionality required by the other parts of Hadoop

The Hadoop File System is just one part of the core components of the Hadoop platform

Data Analysis and Programming Languages

In addition to the algorithms already presented, data analysis also requires implementation and integration into existing applications. At the very least, the algorithms (as mathematical definitions) must be converted into executable code. In the following, some important programming languages are presented for this purpose, as well as frameworks that simplify the work, since these bring along a collection of specialized components (the complete implementation of the algorithms described above, for example).

Python

Python is a programming language that, unlike other programming languages such as C, Fortran, or Java, makes it easier to solve domain problems rather than dealing with the complexity of how a computer works. Python achieves this goal by having the following attributes:

- Python is a high-level language, which means it abstracts away the underlying computer-related technical details. For example, Python does not make its users think too much about managing computer memory or declaring variables correctly and uses safe assumptions about what the programmer is trying to convey. In addition, a high-level language can be expressed in a way that resembles familiar prose or mathematical equations. Python is well suited for beginners because of its ease of understanding and similarity to well-known programming languages, such as Java.
- Python is a general-purpose language, which means that it can be used for all problems - rather than specializing in a particular area, such as statistical analysis. For example, Python can be used to implement artificial intelligence methods as well as statistical analysis.
- Python is an interpreted language, which means that evaluating code to get results can be done immediately, rather than having to go through a time-consuming compiled and executed cycle.
- Python has a standard library and numerous third-party libraries that provide a variety of existing codebases and examples for problem-solving.
- Python has a huge following, which means programmers can quickly find solutions and sample code for problems using Google and Stackoverflow.

In addition, Python has a rich ecosystem for scientific inquiry in the form of many proven and popular open-source packages, including:

- numpy, a Python package for scientific computing,
- matplotlib, a plotting library that produces publication quality figures,
- cartography, a library of cartographic tools for Python,
- netcdf4-python, a Python interface to the netCDF-C library.

R

The "R Foundation" describes R as "a language and environment for statistical computing and graphics." Open-source R was developed by Ross Ihaka and Robert Gentleman at the University of Auckland in New Zealand in the 1990s as a statistical platform for their students and has been extended over the decades with thousands of custom libraries.

- R is data analysis software: Data scientists, statisticians, and analysts - anyone who needs to understand data can use R for statistical analysis, data visualization, and predictive modeling.
- R is a programming language: As an object-oriented language developed by statisticians, R provides objects, operators, and functions that allow users to explore, model, and visualize data.
- R is an environment for statistical analysis: Standard statistical methods are easy to implement in R, and since much of the cutting-edge research in statistics and predictive modeling is done in R, newly developed techniques are often available in R first.
- R is an open-source software project: R is free and has a high standard of quality and numerical accuracy thanks to years of testing and tinkering by users and developers. R's open interfaces allow integration with other applications and systems.
- R has a large community: The R project leadership has grown to more than 20 leading statisticians and computer scientists from around the world, and thousands of contributors have created add-on packages. With two million users, R has a vibrant online community.

R is not only used by academic users, but many large companies also use the R programming language, including Uber, Google, Airbnb, Facebook, and so on. The primary application of R remains statistics, visualization, and machine learning. Also, R has a rich ecosystem for scientific research in the form of many proven and popular open-source packages, including:

1. **sqldf** allows you to perform SQL queries on R data frames. If you have basic SQL knowledge, you can use sqldf to process data very quickly.
2. **forecast** facilitates the fitting of time series models.
3. **plyr** provides a handful of functions that split a data structure into groups, apply a function to each group, and return the results in a data structure.
4. **stringr** offers a number of string operators.
5. **Database drivers** (e.g. RMongo, SQLite, RMySQL) allow R to access the database you are using, saving you time and effort in copying and pasting.
6. **lubridate** facilitates the work with date and time.
7. **ggplot2** makes it easy to create fancy charts.
8. **qcc** is a library for statistical quality control.
9. **reshape2**, as the name suggests, restructures data: It converts data from large format to long format and vice versa.

10. **randomForest** is a machine learning package that enables supervised or unsupervised learning.

Unlike Python, however, R is not suitable for other applications and implementations besides the static core.

SQL

SQL (Structured Query Language) is a standard database language used to create, maintain and retrieve relational databases. SQL was created in the 1970s and has become a very important tool in any data scientist's toolbox because it is crucial for accessing, updating, inserting, manipulating, and modifying data. SQL is used in communicating with relational databases to retrieve the records from the databases.

- **Easy to learn and use** - Unlike other programming languages that require a high level of conceptual understanding and knowledge of the steps required to perform a task, SQL is known for its simplicity through the use of declarative statements. It uses a simple language structure with English words that are easy to understand. If you are a beginner in programming and data science, SQL is one of the best languages to start with. The short syntax makes it possible to query data and gain insights from it. As a data scientist, you definitely need to learn SQL as it is easy to master and required for most projects.
- **Easy to understand and visualize** - SQL helps to explore the dataset sufficiently, visualize, identify the structure and learn what the dataset actually looks like. This helps to identify if there are any missing values, outliers, NULL values and identify the format of the dataset for further use. By slicing, filtering, aggregating, and sorting, SQL allows you to deal with the dataset.
- **Integrates with other languages** - As SQL is powerful in terms of data access, query, and manipulation, it is limited in some aspects like visualization or use of algorithms. SQL integrates well with other scripting languages like R and Python.
- **Big data management** - Most of the time, data science deals with large amounts of data stored in relational databases. Working with such data sets requires high-level solutions to manage them differently from the usual spreadsheets. As the volume of data sets increases, it becomes difficult to use traditional spreadsheets, for example. The best solution for handling large volumes of data is SQL. SQL is capable of managing such data sets.

Scala

Scala is a general-purpose, high-level, multi-paradigm programming language. It is a pure object-oriented programming language that also supports the functional programming approach. There is no primitive data, as everything is an object in Scala. Scala is designed to express common programming patterns in a refined, concise, and type-safe manner. Scala programs can be converted to bytecodes and can run on the JVM (Java Virtual

Machine). Scala is an acronym for “scalable language”, which emphasis the focus of the initiators. It also provides the Javascript runtimes. Scala is heavily influenced by Java and some other programming languages. Scala offers many reasons why it is popular among programmers:

- **Easy**: Scala is a high-level language that is closer to other popular programming languages like Java, C, or C++. So it becomes very easy for anyone to learn Scala. For Java programmers, Scala is easier to learn.
- **Feature richness**: Scala incorporates the features of several languages such as C, C++, or Java, making the language more useful, scalable, and productive.
- **Integration with Java**: The source code of Scala is designed in such a way that the compiler can interpret the Java classes. Also, the associated compiler can use the frameworks, Java libraries, and tools, etc. After compilation, Scala programs can run on the Java Virtual Machine (JVM).
- **Web-based & desktop application development:** For web applications, it provides support by compiling JavaScript. Similar to desktop applications, it can be compiled to JVM bytecode.
- **Used by large companies**: Most of the largest technology companies use Scala. The reason is that it is highly scalable and can be used in the backend.

Julia

Julia was created in 2009 and introduced to the public in 2012. Julia is intended to address the shortcomings of Python and other scientific computing and data processing languages and applications.

- **Julia is compiled and not interpreted**. For faster runtime performance, Julia is compiled just-in-time (JIT) using the LLVM compiler framework. Julia can best approach or match the speed of C.
- **Julia is interactive.** Julia contains a REPL (read-eval-print loop) or interactive command line, similar to Python. Fast, one-time scripts and commands can be entered and executed directly.
- **Julia has a simple syntax**. Julia‘s syntax is similar to Python‘s.
- **Julia combines the advantages of dynamic typing and static typing**. You can specify types for variables, such as “unsigned 32-bit integer”. But you can also create hierarchies of types to allow general cases for handling variables of certain types - for example, to write a function that accepts integers without specifying the length or sign of the integer.
- **Julia can call Python, C, and Fortran libraries**. Julia can work directly with external libraries written in C and Fortran. It is also possible to work with Python code via the PyCall library and even exchange data between Python and Julia.
- **Julia supports metaprogramming**. Julia programs can generate other Julia programs and even modify their code, in a way reminiscent of languages like Lisp.

- **Julia has a full-featured debugger**. Julia 1.1 introduced a debugging suite that runs code in a local REPL and allows you to step through the results, inspect variables, and add breakpoints in the code.

AI Frameworks

Open and free open source software for AI allows anyone to get on the AI bandwagon without spending a lot of time and large resources building the infrastructure. The term open-source software refers to a tool with a source code that is available for free over the internet. For a company that has just launched its first ML initiative, using open source tools can be a great way to practice data science for free before opting for enterprise-level tools like Microsoft Azure or Amazon Machine Learning. But the benefits of using open source tools do not end at availability. Generally, such projects have a large community of developers and data scientists interested in sharing datasets and pre-trained models. For example, instead of building image recognition from scratch, one can use classification models trained on ImageNet's[1] data or create their own from these datasets. With open-source ML tools, one can also use transfer learning, i.e., solve machine learning problems by applying knowledge gained from working on a problem from a related or even distant domain. For example, one can transfer some capacities from the model that has learned to recognize cars to the model that aims to recognize trucks.

Depending on the task to work with, pre-trained models and open datasets may not be as accurate as custom ones but they save a lot of effort and time and do not require you to collect datasets yourself first. According to Andrew Ng, former chief scientist at Baidu and professor at Stanford, the concept of reusing open-source models and datasets will be the second biggest driver of commercial ML success after supervised learning [3].

Among many active and less popular open-source tools, five are selected and presented below.

Tensorflow

Originally developed by Google for internal use, TensorFlow [4] was released under an Apache 2.0 open source license in 2015. The library continues to be used by Google for a number of services, such as speech recognition, photo search, and automatic replies for Gmail inboxes. Google's reputation and the flowchart paradigm used to create models have attracted a large number of contributors to TensorFlow. This has led to public access with detailed documentation and tutorials that provide an easy entry into the world of neural network applications. TensorFlow is a Python tool for both deep neural network exploration and complex mathematical computation, and can even support reinforcement

[1] ImageNet is a free image database which is used for research projects. Each image is additionally assigned to a noun. See: http://www.image-net.org

learning. TensorFlow's uniqueness also lies in its dataflow graph structures, which consist of nodes (mathematical operations) and edges (numerical arrays or tensors).

Datasets and models - The flexibility of TensorFlow is based on the ability to use it for both research and recurrent machine learning tasks. Thus, one can use the low-level API called TensorFlow Core. TensorFlow Core allows you to gain full control over the models and train them with your dataset. However, there are also public and official pre-trained models to develop higher-level APIs based on TensorFlow Core. Some of the most popular models include MNIST,[2] a traditional dataset that helps identify handwritten digits on an image, or Medicare Data, a dataset from Google that is used to predict charges for medical services, among other things.

Audience - For someone looking to use machine learning for the first time, the variety of features in TensorFlow can be a bit overwhelming. Some even argue that the library is not trying to speed up a machine learning curve but make it even steeper. TensorFlow is a low-level library that requires extensive code writing skills and a good understanding of data science specifics to successfully work with the product. Therefore, Tensorflow may not be the first choice if the data science team is IT-centric: there are simpler alternatives for that, which we will discuss.

Use cases - Given the complexity of TensorFlow, use cases mostly involve solutions from large companies with access to machine learning specialists. For example, UK online supermarket Ocado [5] used TensorFlow to prioritize emails arriving at its customer center and improve demand forecasting. Global insurance company Axa [6] also uses the library to predict major claims among its customers.

Theano

Theano is a low-level scientific computing library based on Python that is used for deep learning tasks related to defining, optimizing, and evaluating mathematical expressions. Although it has impressive computational power, users complain about an inaccessible interface and unhelpful error messages. For these reasons, Theano is mainly used in combination with more user-friendly wrappers such as Keras, Lasagne, and Blocks - three high-level frameworks for rapid prototyping and model testing.

Datasets and models - There are public models for Theano but any framework used beyond that also has many tutorials and pre-trained datasets to choose from. Keras, for example, stores available models and detailed tutorials in its documentation.

Audience - Using Lasagna or Keras as a high-level wrapper with Theano, you in turn have access to a variety of tutorials and pre-trained datasets. In addition, Keras is considered one of the easiest libraries to start with in the early stages of deep learning exploration. Since TensorFlow was designed as a replacement for Theano, a large part of its fanbase has left. But there are still many advantages that many data scientists find compelling enough to work with Theano. The simplicity and maturity of Theano alone are important points to consider when making this decision.

[2] See: http://yann.lecun.com/exdb/mnist/.

Use cases - Theano is considered as one industry standard for deep learning research and development, and was originally developed for implementing state-of-the-art deep learning algorithms. However, since people are unlikely to use Theano directly, its many uses expand as it is used as a foundation for other libraries: digital and image recognition, object localization, and even chatbots.

Torch

Torch is often referred to as the easiest deep learning tool for beginners. It has a simple scripting language, Lua, and a helpful community that offers an impressive selection of tutorials and packages for almost any deep learning purpose. Although the underlying language, Lua, is a less common language, Torch itself is widely used - Facebook, Google, and Twitter are known to use it in their AI projects.

Datasets and models - A list of popular datasets loaded for use in Torch can be found on the GitHub cheatsheet page.[3] In addition, Facebook has released official code for implementing Deep Residual Networks (ResNets) with pre-trained models with instructions for fine-tuning your datasets.[4]

Target audience - Regardless of the differences and similarities, the choice will always depend on the underlying language because the availability of experienced Lua developers will always be smaller than that of Python. However, Lua is significantly easier to read, which is reflected in Torch's simple syntax. Torch's active contributors swear by Lua, making it a framework of choice for beginners and those looking to expand their toolset.

Use cases - Facebook uses Torch to create DeepText, a tool that categorizes minute-by-minute text posts shared on the site and provides more personalized content targeting. Twitter was able to use Torch to recommend posts based on an algorithmic timeline (instead of reverse chronological order).

Scikit-Learn

Scikit-learn is a framework designed for supervised and unsupervised machine learning algorithms. As one of the components of the Python scientific ecosystem, it builds on NumPy and SciPy libraries, each responsible for lower-level data science tasks. While NumPy sits on top of Python and deals with numerical computation, SciPy covers more specific numerical routines, such as optimization and interpolation. Scikit-learn was developed specifically for machine learning.

Datasets and models - The library already contains some standard datasets for classification and regression. This is useful for beginners, although the datasets are too small to represent real-world situations. However, the diabetes dataset[5] for measuring disease progression or the iris dataset for pattern recognition are good for illustrating and learning the behavior of machine learning algorithms in scikit. In addition, the library provides

[3] See: https://github.com/torch/torch7/wiki/Cheatsheet

[4] See: https://github.com/facebook/fb.resnet.torch

[5] See: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html

information on loading datasets from external sources, includes example generators for tasks, such as multiclass classification and decomposition, and provides recommendations for using common datasets.

Audience - Although Scikit-learn is a robust library, it emphasizes usability and documentation. Given its simplicity and numerous well-described examples, it is an accessible tool for non-experts to quickly apply machine learning algorithms. According to testimonials from some software houses, Scikit is well suited for production, which is characterized by limited time and human resources.

Use cases - Scikit-learn has been used by a variety of successful tech companies, such as Spotify, Evernote, or Booking.com for product recommendations and customer service.

Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the staff of the Jupyter[6] project. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name Jupyter comes from the main supported programming languages that the tool supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which can be used to write programs in Python - but there are currently over 100 other kernels that can be used. Originally developed for data science applications in Python, R, and Julia, Jupyter Notebook is suitable for all kinds of projects in a variety of ways:

- **Data visualizations** - Most people's first exposure to Jupyter Notebook is through a data visualization, a shared notebook that involves rendering a dataset as a graph. Jupyter Notebook allows you to perform visualizations but also share them, and make interactive changes to the shared code and dataset.
- **Code sharing -** Cloud services like GitHub and Pastebin offer ways to share code but are largely non-interactive. With a Jupyter Notebook, you can view code, execute it, and view the results directly in your web browser.
- **Live code interactions** - Jupyter Notebook code is not static; it can be edited and re-executed incrementally in real-time, with feedback directly in the browser. Notebooks can also embed user controls (e.g., sliders or text entry fields) that can be used as input points for code.
- **Documenting code examples** - If you have a piece of code and want to explain it line by line with live feedback all the way, you can embed it in a Jupyter Notebook. The best part is that the code remains fully functional - you can add interactivity along with the explanation and display and narrate at the same time.

Code is usually not just code. Especially in the field of business analytics, code is part of a thought process, a discussion, even an experiment. This is especially true for data

[6] See: https://jupyter.org

analytics but also almost any other application. With Jupyter, you can create a “notation book” that shows the work: the code, the data, the results, along with the explanations and reflections. Data means nothing if you cannot turn it into insights if you cannot explore, share, and discuss it. Data analysis means little if you cannot explore and try someone else’s results. Jupyter is a tool for exploring, sharing, and discussing. A notebook is easy to share. One can save the notebook and send it as an attachment so someone else can open it with Jupyter. One can upload the notebook to a GitHub repository and have others read it there - GitHub automatically renders the notebook to a static web page. GitHub users can download (clone) their copy of the notebook and any supporting files so they can extend your work. You can view the results, change the code, and see what happens.

Business Analytics and Machine Learning as a Service (Cloud Platforms)

For most companies, business analytics, and especially methods of AI and machine learning, approaches the complexity of space rocket engineering. And if you want to build another Netflix recommendation system, it is. But the trend to offer everything as a service has hit this sophisticated area as well. One can start an ML initiative today without a huge investment, which would be the right move if you are new to data science and just want to leverage the simple value-adds.

One of the most inspiring stories, though a little patheticness from Google’s marketing department certainly has to be taken into account here, is the one about a Japanese farmer[7] who decided to automatically sort cucumbers to help his parents with this tedious task. Unlike the stories that exist about big companies, the young farmer had neither machine learning expertise nor a big budget. But he managed to familiarize himself with TensorFlow and used machine learning to recognize different classes of cucumbers.

By using cloud services, the initial working models can begin to be built and valuable insights can be gained from predictions with a relatively small team. We have already talked about the machine learning strategy. If we now take a look at the best platforms and services on the market, we can at least distinguish between offering artificial intelligence as a service, in the form of machine learning algorithms, and data management and storage.

Machine Learning as a Service (MLaaS)

Machine Learning as a Service (MLaaS) is an umbrella definition of various cloud-based platforms that cover most infrastructure topics, such as data preprocessing, model training and model evaluation, and operational execution. Predictive results can be linked to existing internal IT infrastructure via REST APIs. In this regard, Amazon Machine Learning Services, Azure Machine Learning, Google Cloud AI, SAP Cloud Platform, and

[7] The full story can be found here: https://cloud.google.com/blog/products/gcp/how-a-japanese-cucumber-farmer-is-using-deep-learning-and-tensorflow

IBM Watson are the leading MLaaS services that enable rapid model building and deployment. These should be considered first when assembling a dedicated data science team of available software engineers.

This chapter first provides an overview of the major machine learning as a service platform from Amazon, Google, Microsoft, SAP, and IBM, and then compares the machine learning APIs supported by these vendors.

> Please note that this overview is not intended to provide detailed guidance on when and how to use these platforms. In contrast to the rather concrete selection guide for problems and suitable algorithms before, this chapter rather serves as a rough overview and framework for deciding for or against an approach and a platform. However, as you will see, most platforms are nearly equal and the choice should rather be made based on features not listed here (such as existing contracts, costs, knowledge, etc.).

Data Storage as a Service (DSaaS)
Storage as a Service (SaaS) is a business model in which a large company leases shares in its storage infrastructure to a smaller company or individual. Originally, SaaS vendors focused on secondary storage applications by positioning SaaS as a convenient way to manage backups. But in the meantime, and much more critical to this book, are the current offerings from the major providers that provide databases, data warehouses, data lakes, or streaming services - referred to here as data storage as a service (DSaaS). This means that the first (left) part of the BA framework can then also be created completely externally by cloud-based services, and the rapid availability and low entry hurdle (see the comments above) are thus also valid for data collection, preparation, and storage.

Amazon AWS

Amazon Web Services (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms for individuals, businesses, and public institutions on a subscription basis. The technology allows users to have virtual computer clusters that are accessible over the internet. AWS virtual computers emulate most of the attributes of a real computer, including hardware (CPU(s) & GPU(s), main memory, hard drive/SSD storage); a choice of operating systems; networking; and pre-installed application software, such as web servers, databases, CRM, etc. Each AWS system also virtualizes its console I/O (keyboard, display, and mouse) so that AWS subscribers can connect to their AWS system using any web browser. The browser acts as a window to the virtual computer, allowing subscribers to log in, configure, and use their virtual systems just as they would with a real physical computer. It also provides a variety of services, utilities, and features that can be used independently of an entire server.

AWS technology is deployed on server farms around the world and operated by the Amazon subsidiary. Fees are based on a combination of usage, customer-selected hardware/OS/software/networking features, required availability, redundancy, security, and service options. Subscribers can pay for a single AWS virtual machine, a dedicated physical machine, or clusters of both. As part of the subscription agreement,[8] Amazon provides security for the subscribers' system. AWS operates from many global geographic regions, including 6 in North America.[9]

In 2017, AWS included more than 90 services covering a broad spectrum, including compute, storage, networking, databases, analytics, application services, provisioning, management, mobile devices, developer tools, and tools for IoT and blockchain. Among the most popular are Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3). Most services are not directly accessible to end users but provide functionality via APIs for developers to use in their applications.

The AWS services relevant for BA and AI comprise services at different levels of abstraction and thus each with a different target group. The portfolio has become so extensive that only the most relevant services are presented here, with an attempt to provide an even more comprehensive overview at the end of the chapter.

Amazon's machine learning services are available in two ways: predictive analytics with Amazon ML and the SageMaker tool for data scientists.

Amazon AWS Data Services

The best-known and most used data service of Amazon AWS is certainly Amazon Simple Storage Service (S3). However, fully operated databases (from Amazon's own DynamoDB to many well-known relational databases from MySQl to PostgreSQL) are also offered via Amazon AWD RDS (see Tables 3.4 and 3.5).

Amazon S3 (Data Service)

Amazon Simple Storage Service (Amazon S3) is a scalable, web-based cloud storage service designed to store and archive data and applications on Amazon Web Services. Amazon S3 is designed with a minimal feature set to make it easier for developers to scale storage-intensive applications (storing images or videos). Amazon S3 is an object storage service, which is different from block and file storage. Each object is stored as a file with its metadata and is assigned an ID number. Applications use this ID number to access an object. Unlike file and block storage, a developer can access an object using a REST API.

The S3 cloud storage service allows a subscriber to access the same systems that Amazon uses to power its websites. S3 allows customers to upload, store, and download virtually any file or object up to five terabytes (TB) in size, with the largest single upload limited to five gigabytes (GB). S3 provides 99.9999999999%[10] availability for objects stored in the service and supports multiple security and compliance certifications. There is an extensive partner network of providers that connect their services directly to S3. Data

[8] https://aws.amazon.com/de/agreement/

[9] https://aws.amazon.com/de/about-aws/global-infrastructure/

[10] See: https://aws.amazon.com/de/s3/faqs/.

Table 3.4 Overview of Amazon AWS data services

Amazon Aurora	Managed relational database
Amazon DynamoDB	Managed NoSQL database
Amazon DocumentDB	Fully managed document database (with MongoDB compatibility)
Amazon ElastiCache	In-memory caching system
Amazon Neptune	Fully managed graph database service
Amazon Quantum Ledger Database (QLDB)	Fully managed ledger database
Amazon RDS	Managed relational database service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB
Amazon RDS on VMware	Automatic local database management
Amazon Redshift	Fast and easy data warehousing
Amazon Simple Storage Service (S3)	An object storage service that provides scalability, data availability, security, and performance
Amazon Timestream	Fully managed time-series database
AWS Database Migration Service	Database migration with minimal downtime

Amazon AWS offers a wealth of different and sometimes redundant data services

can be transferred to S3 over the internet via access to S3 APIs. There is also “Amazon S3 Transfer Acceleration,” which are tools for faster uploads and downloads over long distances, and AWS Direct Connect for a private, consistent connection between S3 and the on-premises data center. An administrator can also use “AWS Snowball,”[11] a physical transfer device, to send large amounts of data from an enterprise data center directly to AWS, which then uploads the data to the S3 cloud. In addition, users can integrate other AWS services with S3. For example, an analyst can drop and query data directly on S3 using either Amazon Athena for ad hoc queries or Amazon Redshift Spectrum for more complex analytics.

Working with “AWS Buckets” - Amazon S3 Buckets, which are similar to file folders, store objects consisting of data and its descriptive metadata. An S3 customer first creates a bucket in the AWS region of their choice and gives it a globally unique name. AWS recommends that customers choose regions near them to reduce latency and cost. After creating the bucket, the user then selects a tier for the data, with different S3 tiers with different levels of redundancy, pricing, and accessibility. A bucket can store objects from different S3 storage tiers. Then, the user can set access rights for the objects stored in a bucket through mechanisms such as the AWS Identity and Access Management Service, bucket policies, and access control lists. An AWS customer can interact with an Amazon S3 bucket through the AWS Management Console, the AWS Command Line Interface, or the Application Programming Interfaces (APIs).

[11] See: https://aws.amazon.com/de/snowball/.

Table 3.5 Overview of Amazon AWS ML Services

Amazon SageMaker	Build, train, and deploy custom machine learning models
Amazon Elastic Inference	Acceleration of Deep Learning Inference
Amazon Forecast	Increase forecast accuracy using machine learning
Amazon Lex	Create voice and text chatbots
Amazon Personalize	Integrates real-time recommendations into existing applications
Amazon Polly	Turning text into natural language
Amazon Rekognition	Analyze images and videos
Amazon SageMaker Ground Truth	Create accurate ML training records
Amazon Textract	Extract text and data from documents
Amazon Translate	Natural sounding, fluent translations
Amazon Transcribe	Automatic speech recognition
AWS Deep Learning AMIs	Deep Learning on Amazon EC2
AWS Deep Learning Container	Docker images for Deep Learning
AWS DeepLens	Video camera enabled for Deep Learning
AWS DeepRacer	Autonomous racing car steered by ML on a scale of 1:18
AWS Inferentia	Machine learning inference chip
Apache MXNet in AWS	Scalable, open-source deep learning framework
TensorFlow on AWS	Open-source machine intelligence library
Amazon Elastic Inference	Acceleration of Deep Learning Inference
Amazon Forecast	Increasing forecast accuracy with the help of machine learning

Amazon AWS offers a wealth of different and sometimes redundant ML services

AWS provides several features for Amazon S3 buckets. A user can enable versioning for S3 buckets to preserve every version of an object when an operation is performed on it, such as a copy or delete operation. This helps an IT team prevent the accidental deletion of an object. Similarly, when creating buckets, a user can set up server access logs, object-level API logs, tags, and encryption.

Amazon does not impose a limit on the number of items a subscriber can store; however, there are Amazon S3 Bucket limits. An Amazon S3 Bucket exists within a specific region of the cloud. An AWS customer can use an Amazon S3 API to upload items to a specific bucket. Customers can configure and manage S3 buckets.

Amazon RDS (Data Service)

Amazon Relational Database Service (Amazon RDS) is a managed SQL database service provided by Amazon Web Services (AWS). Amazon RDS supports a range of database types for storing and organizing data and helps with database management tasks, such as migration, backup, recovery, and patching. A cloud administrator uses Amazon RDS to set up, manage, and scale a relational database instance in the cloud. The service also automatically backs up RDS database instances, takes a daily snapshot of the data, and stores transaction logs to enable point-in-time recovery. RDS also automatically patches the database engine software.

To increase the availability and reliability of production workloads, Amazon RDS enables replication. An administrator can also enable automatic failover across multiple availability zones with synchronous data replication. An AWS user controls Amazon RDS through the AWS Management Console, Amazon RDS APIs, or the AWS Command Line Interface. A database administrator can create, configure, manage, and delete an Amazon RDS instance, a cloud database environment, and the compute and storage resources it uses. Depending on which database engine an admin chooses, he or she can create multiple databases or schemas. Amazon RDS limits each customer to a total of 40 database instances per account. Amazon RDS provides support for major and minor versions of database models over time, and an admin can specify the desired version when creating a database instance. In most cases, Amazon RDS can support developer code, applications, and tools already in use with existing databases. AWS users can launch six types of database engines within Amazon RDS:

Amazon Aurora A custom implementation of the popular MySQL RDBMS that AWS claims are designed for the cloud by combining storage, networking, compute, systems, and database software into a single service. Aurora automatically scales in 10GB increments as usage increases without taking applications offline. Database performance scales linearly based on capacity and can result in higher input/output operations per second (IOPS) as needed. To ensure high availability, Aurora automatically replicates across three availability zones (AZs), each with two copies of the data. This redundancy allows Aurora to process data as soon as at least four of six writes complete, rather than waiting for all writes to complete. Aurora DBs can create up to 15 replicas with fast 10- to 20-millisecond failover between them.

MySQL This service provides a default installation of MySQL Community Edition using InnoDB as the default storage engine. Users can configure MySQL versions of 5.5, 5.6, and 5.7 - and minor versions for each - during setup. Amazon has committed to supporting versions for at least three years after the initial release on RDS. Unlike Aurora, standard MySQL instances do not automatically scale capacity or replicate across multiple AZs. In addition, the IT team pays for allocated storage, whereas Aurora users pay only for the resources the database uses. MySQL has a wider scaling range and can be provisioned with tiny t2.micro instances, while r3.large is the smallest for Aurora.

MariaDB This popular offshoot of the MySQL database provides instances with features similar to the RDS MySQL service, including automatic patching and database snapshots. Instances can use either magnetic general-purpose SSDs or provisioned SSD Elastic Block Store storage, supporting from 1000 to 30,000 IOPS per instance. Developers can automatically replicate instances across AZs, with automatic failover from the primary database.

Oracle Enterprises can deploy this database as either on-demand or reserved DB instances. As with EC2 instances, reserved databases charge an upfront fee with a 20% to 60% discount on hourly usage, depending on the terms of the reservation. Instances have an Oracle license but customers can also bring their own, subject to Oracle restrictions on cloud usage.

Microsoft SQL Server Similar to Oracle, this option is a default instance of the Windows database. Amazon RDS supports SQL Server 2008 R2, 2012, and 2014 for Express, Web, and Standard editions, and Enterprise 2008 R2 and 2012 for Enterprise. Customers can reserve instances. Organizations with Microsoft Software Assurance can bring their licenses for significant discounts on base usage as needed.

PostgreSQL This service provides the same core management and patching features described for the other open source databases, at about the same price. Because of the availability of its add-on modules, PostgreSQL is popular with developers who create geospatial, statistical, and machine-readable backends so they can move existing application code into the RDS with little modification.

Amazon AWS ML Services

Amazon Machine Learning for predictive analytics is one of the most automated solutions on the market and the best solution for deadline-critical operations. The service can load data from various sources, including Amazon RDS, Amazon Redshift, CSV files, etc. All data pre-processing operations are performed automatically: the service identifies which fields are categorical and which are numeric, and does not prompt the user to choose the methods of further data preprocessing (dimension reduction and whitening).

Amazon ML's prediction capabilities are limited to three options: binary classification, multiclass classification, and regression. However, this Amazon ML service does not support unsupervised learning methods and a user needs to select a target variable to label it in a training set. Also, a user does not need to know any machine learning methods as Amazon automatically selects them after looking at the provided data.

This high level of automation works as both an advantage and a disadvantage for using Amazon ML. If you need a fully automated yet limited solution, the service can meet expectations - if not, there is Amazon SageMaker.

Amazon SageMaker (ML Service)

SageMaker is a machine learning environment designed to simplify the work of a data scientist by providing tools for rapid model building and deployment. Amazon also has built-in algorithms optimized for large data sets and computation in distributed systems. These currently include[12]:

- Discrete, binary, and multiple-choice classification
- DeepAR forecasting algorithm
- K-means algorithm
- K-nearest-neighbours (k-NN) algorithm
- Latent Dirichlet Allocation (LDA) algorithm
- Linear-learner algorithm

[12] More here: https://docs.aws.amazon.com/sagemaker/latest/dg/algos.html

- Random Cut Forest (RCF) algorithm
- XGBoost algorithm
- BlazingText algorithm
- Image classification Algorithm
- IP insights algorithm
- Neural Topic Model (NTM) algorithm
- Object2Vec algorithm
- Object detection algorithm
- Principal Component Analysis (PCA) algorithm
- Semantic segmentation algorithm
- Sequence-to-sequence algorithm

Amazon SageMaker also supports Jupyter Notebooks (section "Jupyter Notebook"), which helps developers create and share live code. For SageMaker users, these notebooks include drivers, packages, and libraries for common deep learning platforms and frameworks. A developer can launch a preconfigured notebook that AWS delivers for a variety of uses and use cases, and then customize it to the dataset and schema that the developer wants to train. Developers can also use custom algorithms written in any of the supported ML frameworks or any code packaged as a Docker container image. SageMaker can pull data from Amazon Simple Storage Service (S3) and there is no practical limit on the size of the dataset.

SageMaker can use training jobs to set parameters for the model, deploy the final model, and validate the production model with tests from a known dataset. The service can also use custom TensorFlow or MXNet code to train a machine learning model. Although the included Python libraries are suitable for most situations, developers can also use SageMaker with Spark on Amazon EMR for more advanced data processing needs. Developers can run a variety of applications that use SageMaker, including targeted marketing for ads or promotions, fraud protection, credit scoring, predictive maintenance via the Internet of Things (predictive maintenance), data, and other time series forecasting.

Amazon Forecast (MLaaS)

Amazon Forecast (see the overview in Fig. 3.3) is a fully managed service that uses machine learning to provide highly accurate forecasts. Previously used tools such as spreadsheets create forecasts by looking at a historical data series called time-series data. For example, such tools may try to predict future sales of a raincoat by looking only at past sales data, assuming that the future is determined by the past. This approach can make it difficult to make accurate predictions for large data sets with irregular trends. It is also not easy to combine data series that change over time (such as price, discounts, web traffic, and a number of employees) with relevant independent variables, such as product features and locations. Based on the same technology as Amazon itself, Amazon Forecast uses machine learning to combine time series data with additional variables to create forecasts. Amazon Forecast does not require any machine learning experience to use. All you have

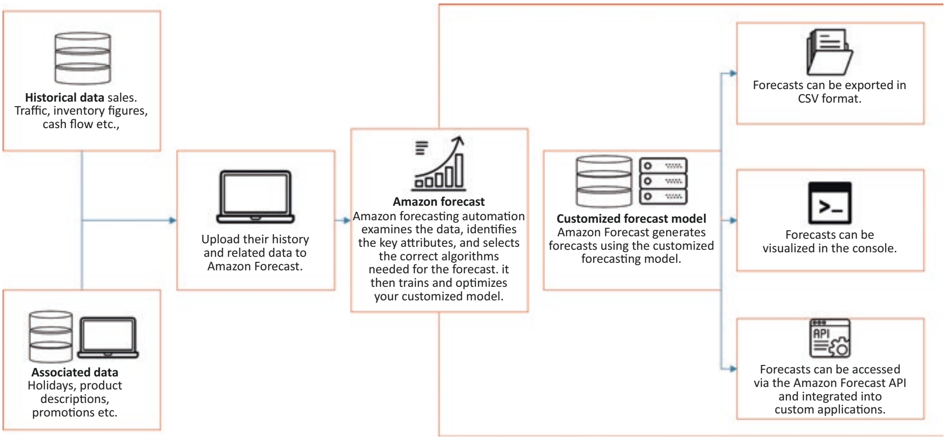

Historical data sales. Traffic, inventory figures, cash flow etc.,
Associated data Holidays, product descriptions, promotions etc.
Upload their history and related data to Amazon Forecast.
Amazon forecast Amazon forecasting automation examines the data, identifies the key attributes, and selects the correct algorithms needed for the forecast. it then trains and optimizes your customized model.
Customized forecast model Amazon Forecast generates forecasts using the customized forecasting model.
Forecasts can be exported in CSV format.
Forecasts can be visualized in the console.
API
Forecasts can be accessed via the Amazon Forecast API and integrated into custom applications.

Fig. 3.3 Schematic representation of the Amazon Forecast Service

to do is provide historical data as well as additional data that you think might affect the forecasts. For example, demand for a particular color of shirt may change with the seasons and the location of the store. This complex relationship is difficult to determine manually but machine learning is ideally suited to identify it. Once the data is provided, Amazon Forecast automatically examines it, identifies what makes sense, and creates a forecasting model capable of making predictions that are up to 50% more accurate than looking at time series data alone.

For example, you can use Amazon Forecast to create the following forecasts:

- Demand for retail products, such as demand for products sold on a website or in a particular store or location (for sales planning or even reordering purposes).
- Supply chain demand, including the number of raw materials, services, or other inputs needed to produce the products.
- Resource requirements, such as the number of call center agents, contract workers, IT staff, and/or amount of energy needed to meet the demand.
- Operational metrics, such as web traffic, AWS usage, or IoT sensor usage.
- Key business figures, such as cash flow, revenue, profits, and expenses, by region or service

Amazon Forecast is a fully managed service, so there are no servers to provision and no machine learning models to create, train, or deploy. Users only pay for what they use, and there are no minimum fees or upfront commitments.

Amazon Forecast greatly simplifies the creation of machine learning models. In addition to providing a set of predefined algorithms, Forecast provides an AutoML option for model training. AutoML automates complex machine learning tasks, such as algorithm selection, hyperparameter setting, iterative modeling, and model evaluation. Developers

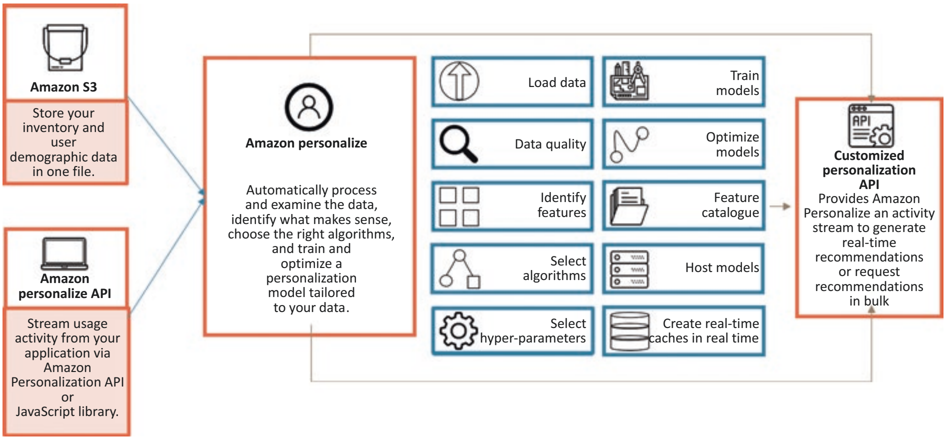

Amazon S3
Store your inventory and user demographic data in one file.
Amazon personalize API
Stream usage activity from your application via Amazon Personalization API or JavaScript library.
Amazon personalize
Automatically process and examine the data, identify what makes sense, choose the right algorithms, and train and optimize a personalization model tailored to your data.
Load data
Data quality
Identify features
Select algorithms
Select hyper-parameters
Train models
Optimize models
Feature catalogue
Host models
Create real-time caches in real time
API
Customized personalization API
Provides Amazon Personalize an activity stream to generate real-time recommendations or request recommendations in bulk

Fig. 3.4 Schematic representation Amazon Personalize

without machine learning experience can import training data into one or more Amazon Forecast datasets, train predictors, and generate forecasts using the Amazon Forecast APIs, the AWS Command Line Interface (AWS CLI), or the Amazon Forecast console.

Amazon Forecast offers the following additional benefits over homegrown models:

- Accuracy - Amazon Forecast uses deep neural networks and traditional statistical methods for forecasting. Given much related time series, forecasts made using Amazon Forecast deep learning algorithms, such as DeepAR+[13] and NPTS[14] are typically more accurate than forecasts made using traditional methods, such as exponential smoothing.
- Ease of use - the Amazon Forecasting Console can be used to look up and visualize forecasts for any time series with different granularities. Metrics for the accuracy of your forecasts can also be viewed.

Amazon Personalize (Analytics Service)

Amazon Personalize[15] (see Fig. 3.4 for an overview) is a machine learning service that makes it easy for developers to use personalization in applications and provide customized recommendations to customers. It reflects the vast experience Amazon has in building personalization systems. For example, Amazon Personalize can be used in a variety of scenarios, including recommendations for users based on their preferences and behaviors, personalized re-ranking of results, and personalized content for emails and notifications. Amazon Personalize does not require extensive machine learning experience. Pre-defined

[13] An algorithm developed by Amazon. See: https://docs.aws.amazon.com/forecast/latest/dg/aws-forecast-recipe-deeparplus.html

[14] Also, an algorithm developed by Amazon: https://docs.aws.amazon.com/forecast/latest/dg/aws-forecast-recipe-npts.html

[15] See: https://aws.amazon.com/de/personalize/.

solution variants (a trained Amazon Personalize recommendation model) can be created, trained, and deployed using the AWS Console or programmatically using the AWS SDK. All the developer needs to do is the following:

1. format input data and upload it to an Amazon S3 bucket or send real-time event data,
2. select a training recipe (algorithm) to be applied to the data,
3. train a solution variant,
4. provision of the solution via interface,
5. integration into existing applications.

Amazon Personalize can capture live user events to provide real-time personalization. Amazon Personalize can combine real-time user activity data with existing user profiles and item information to recommend the most relevant items based on the user's current session and activity.

"Predefined Recipes"[16]in Amazon Personalize allow you to create custom personalization models without needing any knowledge of machine learning. The user can choose which model ("recipe") to train a solution version with, or let Amazon Personalize decide which recipe is best for the data. To help decide which recipe to use, Amazon Personalize provides extensive metrics on the performance of a trained solution version.

Amazon Personalize has an AWS console that you can use to create, manage, and deploy solution variants. Alternatively, you can use the AWS Command Line Interface (AWS CLI) or one of the Amazon Personalize SDKs.

Amazon Personalize consists of three related components:

- Amazon Personalize - to create, manage, and deploy solution variants.
- Amazon Personalize Events - to record user events that can be added to your existing training data.
- Amazon Personalize Runtime - to get recommendations from a campaign (provided solution variant).

Google Cloud Platform

The Google Cloud Platform (GCP) is a collection of cloud services offered by Google. The platform includes a set of hosted services for computing, storage, and application development that run on Google hardware. Google Cloud Platform services can be accessed by software developers, cloud administrators, and other enterprise IT professionals over the internet or through a dedicated network connection. In April 2008, Google announced a preview version of App Engine, a developer tool that allows users to run their

[16] See: https://docs.aws.amazon.com/personalize/latest/dg/working-with-predefined-recipes.html. Reprint rights: not necessary notes publisher/setter.

web applications on Google infrastructure. The self-proclaimed goal of App Engine was to “Make it easy to start with a new web app, and then make it easy to scale when that app reaches the point where it’s generating lots of traffic and has millions of users.” [7].

Google Cloud Platform provides services for computing, storage, networking, big data, machine learning, and the Internet of Things (IoT), as well as cloud management, security, and developer tools. The core cloud computing products in Google Cloud Platform include:

- Google Compute Engine, an infrastructure-as-a-service (IaaS) offering that provides users with virtual machine instances.
- Google App Engine, a platform-as-a-service (PaaS) offering that gives software developers access to Google’s scalable hosting. Developers can also use a software developer kit (SDK) to build software products that run on App Engine.
- Google Cloud Storage, a cloud storage platform for storing large, unstructured data sets. Google also offers database storage options, including Cloud Datastore for NoSQL with non-relational storage, Cloud SQL for MySQL with fully relational storage, and Google’s native Cloud Bigtable database.
- Google Container Engine, a management and orchestration system for Docker containers running within Google’s public cloud. Google Container Engine is based on the Google Kubernetes container orchestration engine.

Google continues to add higher-value services to its cloud platforms, such as those related to big data and machine learning. Google’s data services include those for data processing and analytics, such as Google BigQuery for SQL-like queries on multi-terabyte datasets. In addition, Google Cloud Dataflow is a data processing service intended for analytics, extract, transform and load (ETL), and real-time computation projects. The platform also includes Google Cloud Dataproc, which offers Apache Spark and Hadoop services for big data processing. For artificial intelligence, Google offers Cloud Machine Learning Engine, a managed service that allows users to build and train machine learning models. Various APIs are available for translating and analyzing speech, text, images, and videos. Google also offers services for IoT, such as Google Cloud IoT Core, a set of managed services that allow users to consume and manage data from IoT devices. The range of Google Cloud Platform services is constantly evolving, and Google regularly introduces, changes, or discontinues services based on user demand or competitive pressures.

Data Services from Google

An overview of Google’s data services is shown in Table 3.6.

Firebase/Google Firebase Realtime Database

Google Firebase is an application development software supported by Google that allows developers to build iOS, Android, and web applications (see Fig. 3.5 for an overview). Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating

Table 3.6 Overview of Google's data services

Google Cloud SQL	MySQL and PostgreSQL database service
Google Cloud Bigtable	Column-oriented NoSQL database service
Google Cloud Spanner	Service for business-critical, scalable, relational databases
Google Cloud Datastore	NoSQL database service for documents
Google Cloud Memorystore	Fully-managed in-memory data storage service
Google Cloud Firestore	Store data from mobile apps and web applications worldwide
Google Firebase Realtime Database	Store and synchronize data in real-time
Google Cloud SQL	MySQL and PostgreSQL database service

Google Cloud Platform data services overview

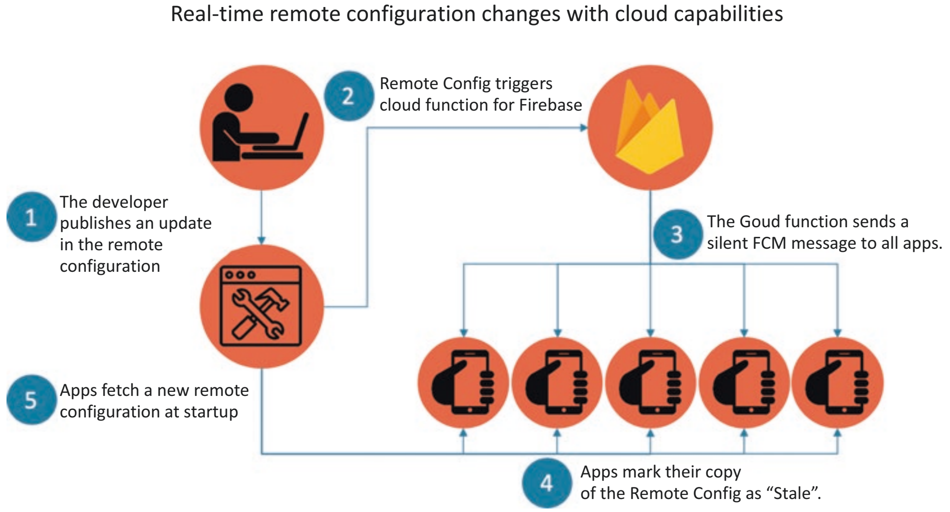

Real-time remote configuration changes with cloud capabilities
2
Remote Config triggers cloud function for Firebase
1
The developer publishes an update in the remote configuration
3
The Goud function sends a silent FCM message to all apps.
5
Apps fetch a new remote configuration at startup
4
Apps mark their copy of the Remote Config as "Stale".

Fig. 3.5 Google Firebase workflow

marketing and product tests. The Firebase real-time database is a cloud-based NoSQL database that allows data to be stored and synchronized between users in real-time. The data is synced in real-time across all clients and is still available even if an app goes offline.

The basics - The Firebase Realtime Database is Firebase's original hosted NoSQL database. This stores data in JavaScript Object Notation (JSON) format and makes it available in real-time. The database is primarily intended for mobile and web app development. Developers can use Android, iOS, and JavaScript SDKs as well as REST APIs. Google Cloud Firestore is also part of the Firebase portfolio and is a flexible, scalable, hosted NoSQL database better suited for demanding complex applications. This newer offering provides some additional features and more functionality compared to the real-time database. For example, like its predecessor, Cloud Firestore is used to build mobile and web applications using SDKs for iOS and Android but it can also be used for server-side development via Node.js, Java, Python, and Go SDKs, as well as REST and RPC APIs. For this reason, Firestore is best suited for newer application development projects.

Data structure - Each NoSQL database stores data in a different way. The Firebase real-time database stores data in a large JSON tree, which is a group of JSON documents. This works well for handling simple data but can fall short if you need a database to organize large amounts of data or manage data hierarchically. Google Cloud Firestore uses documents and they contain fields that correspond to actual values. These documents are then stored as collections and sub-collections that can be organized to collect related data or facilitate database queries. The database can support many types of data - including nested objects and complex hierarchical data structures.

Offline support - Firebase Realtime Database supports an offline mode in which persistence is maintained via a local cache even when the application is disconnected from the internet. The database is automatically updated and synchronized with its instance as soon as the application reconnects to the internet. This allows the application to continue functioning even when network access is slow or intermittent. Google Cloud Firestore also offers an offline mode that caches data for the client and allows apps to continue reading and writing to the database without internet access. However, it also integrates with other Google Cloud services like Cloud Functions as well as open-source libraries. This allows for more versatile actions and flexibility in app design - only changes to the database are shared during synchronization.

Writes and Transactions - Firebase Realtime Database uses simple write and transaction procedures. Data is written through special sets and regular updates. Data is also stored through transactions. Google Cloud Firestore increases flexibility in write and transaction operations, writes data with sets and updates, and allows more complex operations with transformations. Transactions can also be read and written to any part of the database, while Firebase Realtime can only use transactions for a specific data subtree.

Reliability and scalability - In terms of reliability and performance, Firebase Realtime Database offers extremely fast operations, but the database is limited to the availability zones of a single region. Also, the database must be distributed across multiple instances if it exceeds 100,000 concurrent connections and 1000 writes per second. In comparison, Google Cloud Firestore offers the reliability of a native multi-region service. It is also more scalable with current limits of one million concurrent connections and 10,000 writes per second - and this is expected to increase in the future.

Google BigQuery

BigQuery is a data warehouse that processes and analyzes large data sets using SQL queries. These services can capture and explore streaming data for real-time analysis. The warehouse stores data in Google Capacitor’s columnar data format and users can load data via streaming or batch loads. To load, export, query, and copy data, one uses the classic web user interface, the web user interface in the GCP console, the bq command-line tool, or client libraries. Since BigQuery is a serverless offering, companies only pay for the memory and computations they need.

BigQuery is designed to analyze billions of rows of data using SQL-like syntax. It runs on Google's cloud storage infrastructure and can be accessed using a REST-oriented application program interface (API). BigQuery, which was released as V2 in 2011, is what Google calls an "externalized version" of its homegrown query service software Dremel. Dremel and BigQuery use column-oriented storage for fast data scanning and a tree architecture for sending queries and merging results across massive clusters of computers.

BigQuery in its original form was used in Google to track device installation data, generate crash reports, and analyze spam. Since its release, BigQuery features have been continuously improved. In early 2013, data join and timestamps were added to the service. Later, stream data insertion features were also added.

ML Services from Google

Table 3.7 shows an overview of the ML services of the GCP.

Table 3.7 Overview of the ML services of the GCP

Google Hub (Alpha)	Find, share, and deploy AI components in the Google Cloud
Google Cloud AutoML (Beta)	Easily train high-quality custom ML models
Google Cloud TPU	Train and run ML models faster than ever before
Google Cloud Machine Learning Engine	Create first-class models and make them available in production
Google Cloud Talent Solution	Hiring new employees with AI support
Google Dialogflow Enterprise Edition	Implement dialog-oriented communication across devices and platforms
Google Cloud Natural Language	Extract data from unstructured text
Google Cloud Speech-to-Text	ML-assisted conversion of speech to text
Google Cloud Text-to-Speech	ML-assisted conversion of text to speech
Google Cloud Translation	Dynamic translation between languages
Google Cloud Vision	Extract information from images using machine learning
Google Cloud Video Intelligence	Extract metadata from videos
Google Cloud Inference API (Alpha)	Quickly run large correlations in typed timeline datasets
Google Firebase Predictions (Beta)	Intelligently segment users based on predicted behavior
Google Cloud Deep Learning VM Image	Preconfigured VMs for Deep Learning applications

The GCP offers a wealth of different and sometimes redundant ML services

Google Prediction API and Cloud AutoML

Google offers AI services at two levels: a machine learning environment for savvy data scientists and a highly automated Google Prediction API.[17] Google is currently testing Cloud AutoML. The product is currently in beta, so it does not even have documentation yet. The initial information states that AutoML, similar to Amazon SageMaker, will allow people without data science skills to automatically train models on their data. The first product to hit the market is AutoML Vision, which is capable of building custom image recognition models. In the longer term, Google expects to cover even more areas. According to the announcement, the service will support the following areas:

- Image recognition with AutoML Vision and AutoML Video Intelligence.
- Language recognition and translation with AutoML Natural Language and AutoML Translation.
- Data management with AutoML Table.

Google Cloud Machine Learning Engine (Cloud Machine Learning Engine)

High automation of the former Prediction API was only possible at the cost of flexibility. Google ML Engine is the direct opposite. The product is aimed at experienced data scientists, is very flexible, and recommends the use of cloud infrastructure with TensorFlow. In addition, Google is testing several other popular frameworks, such as XGBoost, Scikit-leran, and Keras. So in principle, the ML Engine is pretty similar to SageMaker.

TensorFlow is another Google product that is an open-source machine learning library with various data science tools rather than ML-as-a-service. It does not have a visual interface and the learning curve for TensorFlow would be quite steep. However, the library is also aimed at software developers who are planning to transition to data science. Google TensorFlow is pretty powerful but it is mostly geared towards deep neural network tasks.

Basically, the combination of TensorFlow and Google Cloud Services provides infrastructure-as-a-service and platform-as-a-service solutions under the three-tier model of cloud services.

IBM Watson

IBM offers a single machine learning platform for both experienced data scientists and those new to the industry. Technically, the system offers two approaches: an automated and a manual implementation (with the latter intended for proven experts). Similar to the outdated Google Prediction API or Amazon ML, IBM's Watson Studio has a model builder reminiscent of a fully automated data processing and model creation interface that requires little to no training to get started with data processing, model creation, and deployment to production.

[17] This was discontinued on April 30, 2018, see: https://cloud.google.com/prediction/?hl=de

The automated part can solve three main types of tasks: binary classification, multiclass classification, and regression. One can either choose a fully automated approach or manually select the ML method to use. Currently, IBM has ten methods to cover these three groups of tasks:

- Logistic regression
- Decision trees (Decision tree classifier)
- Random forest classification
- Gradient-boosted tree classification
- Naive Bayes classification
- Linear regression
- Regression-based on decision trees
- Random forest Regression
- Gradient-boosted tree regression
- Isotonic regression

Separately, IBM provides a deep neural network training workflow with a flow editor interface similar to the one used in Azure ML Studio. If you are looking for more advanced features, IBM ML has notebooks like Jupiter to manually program models using popular frameworks like TensorFlow, Scikit-learn, PyTorch, and others.

Microsoft Azure

When it comes to machine learning as a service (MLaaS) platforms, Microsoft's Azure also seems to have a versatile toolset in the MLaaS market. It covers the majority of ML-related tasks, offers two different products for building custom models, and has a solid set of APIs for those who do not want to attack data science with their bare hands. Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It offers a range of cloud services, including services for computing, analytics, storage, and networking. In addition to traditional cloud offerings, such as virtual machines, object storage, and content delivery networks (CDNs), Azure also offers services based on proprietary Microsoft technologies. Azure also offers cloud-hosted versions of popular Microsoft enterprise solutions, such as Active Directory and SQL Server.

Data Services from Microsoft Azure

An overview of the data services on Azure is shown in Table 3.8.

Azure Cosmos DB

Azure Cosmos DB is a cloud database that supports multiple ways to store and process data; as such, it is classified as a multi-model database. In multi-model databases, different

Table 3.8 Overview - Data Services on Azure

Azure Cosmos DB	Globally distributed database with support for multiple data models at any scale
Azure SQL database	Managed relational SQL database as DaaS solution (database-as-a-service)
Azure Database for MySQL	Managed MySQL database service for app developers
Azure Database for PostgreSQL	Managed PostgreSQL database service for app developers
Azure Database for MariaDB	Managed MariaDB database service for app developers
SQL Server on virtual computers	Hosting SQL Server enterprise applications in the cloud
Azure Database Migration Service	Easier migration of local databases to the cloud
Azure Cache for Redis	Top performance for applications thanks to high throughput and low-latency data access
SQL Server Stretch Database	Dynamic stretching of on-premises SQL Server databases to Azure

Microsoft Azure also offers a range of data services, some of which are redundant

database modules are natively supported and made accessible through common APIs. Thus, one is not limited to a single data model as is the case with dedicated graphs, key values, or document repositories, for example.

Azure Cosmos DB grew in part out of Microsoft Research's work to improve data development methods for large-scale applications. This work began in 2010 as "Project Florence"[18] and was commercialized by Microsoft in 2015 with the release of Azure DocumentDB. Azure Cosmos DB, which became generally available in May 2017, is the next generation of document-oriented databases, essentially replacing the document-oriented NoSQL data model. Cosmos DB provides support for key values, graphs, and geospatial data, among others. In this regard, Azure Cosmos DB is only available as a cloud service and features support for global data distribution, i.e., data partitioning across multiple Azure cloud regions or zones (meaning geographically separated Microsoft data centers). Azure Cosmos DB uses containers called "collections" to store data. Without explicit programming, Azure Cosmos DB's global distribution paradigm places data closer to users' physical locations. The database also provides an advanced consistency tuning model that aims to work around issues that data architects actually have to handle to achieve difficult tradeoffs between throughput, space, and consistency in distributed systems, as described in the oft-cited CAP theorem.[19]

[18] See: https://azure.microsoft.com/de-de/blog/dear-documentdb-customers-welcome-to-azure-cosmos-db/.

[19] The theorem states that networked and distributed systems can only guarantee two of the following three properties: consistency, availability, and partition tolerance. See also [8, 9].

Table 3.9 Overview of the ML services of Azure

Azure Cognitive Services	Intelligent API functions for contextual interaction
Azure Bot Services	Intelligent, serverless bot service with demand-driven scaling
Azure Databricks	Fast, simple, and collaborative analytics platform based on Apache Spark
Azure Machine Learning	Create, train, and deploy models - from cloud to edge

Microsoft Azure's ML services are divided into four categories

As with document DB, Azure Cosmos DB allows data developers to work with flexible data schemas that are easier to create and update than the more common relational schemas.

ML Services from Microsoft Azure

Microsoft Azure also offers a comprehensive portfolio of services in the area of machine learning. The Azure services can be roughly divided into two main categories: Azure Machine Learning Studio and Azure Bot Service, with other services (see Table 3.9) available as well. Microsoft's list of machine learning products is similar to Amazon's, but Azure is more flexible in terms of out-of-the-box algorithms offered.

Microsoft Azure Machine Learning Studio

The Azure Machine Learning platform aims to provide a powerful cloud environment for both novices and experienced data scientists. ML Studio is the main MLaaS package in Azure's portfolio. Almost all operations in Azure ML Studio must be performed with a graphical drag-and-drop interface. This includes data exploration, preprocessing, method selection, and validation of modeling results. Approaching machine learning with Azure comes with a bit of a learning curve. But it eventually leads to a deeper understanding of all the important techniques in the field. Azure ML's graphical interface visualizes each step within the workflow and supports newcomers. Perhaps this is also the main advantage of using Azure: the variety of algorithms available. The studio supports around 100 methods that deal with classification (binary and multiple), anomaly detection, regression, recommendation, and text analysis. It is worth noting that the platform has a clustering algorithm (k-means). Another great part of Azure ML is the Cortana Intelligence Gallery. It is a collection of machine learning solutions provided by the community to be reused. The Azure product is a powerful tool to get started with machine learning.

Microsoft Azure Machine Learning Services

In September 2017, Microsoft introduced a new set of ML-focused services that were given the umbrella name Azure Machine Learning Services. The release was responsible for some confusion in the Azure developer community, as users now had to choose between the two different platforms that cannot be integrated. Azure ML Services provides end-to-end lifecycle management that tracks all experiments across the team, storing code, configuration, parameter settings, and environment details to make it easy to evaluate and replicate each experiment across a team. Once there is a preferred model, you can simply encapsulate it in a container and deploy it to Azure, on-prem, or IoT devices, and also scale and manage this with relative ease - as it is "just another container" running on Kubernetes.

Basically, the services propose a supportive environment to create models, experiment with them, and use a variety of open source components and frameworks. Unlike ML Studio, the service does not have built-in methods and requires custom model building. The platform is aimed at more experienced data scientists. For this audience, Azure ML Services offers a powerful toolset to manage ML experiments, use common frameworks, such as TensorFlow, Scikit-learn, etc. (which is not possible with ML Studio, for example), and allows the resulting models to be deployed in production in a third-party service such as Docker.

A brief overview of what the platform has to offer:

- **Python packages** - These proprietary packages have libraries and functions that target four main task groups: computer vision, prediction, text analysis, and hardware acceleration.
- **Experimentation environment** - With all Python tools and frameworks, data scientists can create, version, and compare different models.
- **Model management** - The tool provides an environment to host, version, manage, and monitor models running on Azure, on-premises, or even Edge devices.
- **Workbench** - This product is a convenient desktop and command-line environment with dashboards and evaluation tools for tracking model development.
- **Visual Studio Tools for AI** - Basically, this extension adds tools to the VS IDE to work with Deep Learning and other AI products.

Overview of Other Microsoft Azure Services

Since the services selected above represent only a fraction of Azure's offerings, Table 3.10 briefly describes other relevant services and their intended uses.

SAP Business Technology Platform (SAP BTP)

SAP Business Technology Platform (SAP BTP) (formerly known as SAP Cloud Platform or "SCP") is a platform-as-a-service (PaaS) product that provides a development and runtime environment for cloud applications. Based on SAP HANA in-memory database technology and using open source and open standards, SCP enables independent software vendors, startups, and developers to build and test HANA-based cloud applications. Originally introduced on October 16, 2012, as SAP NetWeaver Cloud[20] from the SAP HANA Cloud Portfolio, the cloud platform was relaunched on May 13, 2013, under the new name SAP HANA Cloud Platform as the foundation for SAP cloud products, including

[20] See: https://blogs.sap.com/2012/06/01/sap-netweaver-cloud-the-road-forward/.

Table 3.10 Other Azure services

Intended use	Description	Solution from Microsoft Azure
Data management	A fully managed, elastic data warehouse	SQL data warehouse
Data management and analysis	A fully managed analytics platform - optimized for Azure - for collaboration based on Apache Spark.	Azure Databricks
Data management and analysis	A fully managed Hadoop and Spark cloud service.	Azure HDInsight
Data integration	A data integration service to orchestrate and automate data moves and transformations	Azure Data Factory
Data modeling	Open and flexible artificial intelligence deployment including Jupyter environment	Azure Machine Learning
Data streaming	Real-time processing of data streams from millions of IoT devices	Azure Stream Analytics
Data analysis (focus on security)	A fully managed, on-demand analytics service with order-based payment and enterprise-grade security, monitoring, and support	Azure Data Lake Analytics
Data analysis	Analysis-as-a-Service with a focus on large companies	Azure Analysis Services
Data streaming	A hyper-scale telemetry data collection service that collects, transforms, and stores millions of events	Azure Event Hubs
Data analysis	Fast and highly scalable service for examining data	Azure Data Explorer

In addition to data and ML services, Azure offers a wide range of other services

SAP BusinessObjects Cloud. The scope of SAP HANA Cloud Platform has been steadily expanding since its launch in 2012, with SAP currently touting over 4000 customers and 500 partners using SAP HANA Cloud Platform. On February 27, 2017, the original SAP HANA Cloud Platform was renamed SAP Cloud Platform at Mobile World Congress. In mid-January 2021, the SAP Cloud Platform brand was officially retired to support SAP's One Platform strategy and rebranded to SAP Business Technology Platform (SAP BTP).

According to SAP, BTP is primarily designed to enable companies to extend existing on-premises or cloud-based (ERP) applications with next-generation technologies, such as building and deploying advanced analytics, blockchain, machine learning, new business clouds, and mobile applications; integrating and connecting enterprise applications regardless of application location or data source; and connecting enterprise applications and data to the IoT. For example, BTP enables the integration of SAP S/4HANA Finance with cloud applications, such as SAP Ariba or SAP SuccessFactors. It can also integrate these applications with non-SAP systems and data sources, including social media sites and third-party enterprise applications. BTP is based on open standards and gives developers flexibility and control over which clouds, frameworks, and applications to use, according to SAP. BTP uses several development environments, including Cloud Foundry and Neo, and offers a variety of programming languages. BTP is available in two commercial models: subscription-based and consumption-based. These options allow companies to flexibly customize BTP services to meet organizational needs, SAP said. Under the subscription model, customers get access to SAP Cloud Platform services for a fixed price and defined time and can use as many services as they want. This model allows companies to handle their IT investments with predictable costs as long as they subscribe to the service. Under the consumption model, customers can purchase BTP services through credits and use them as they see fit. This configuration allows companies to quickly launch and expand development projects as business needs change. BTP credits are paid upfront, and a cloud credit is maintained for all services used. SAP Cloud Platform offers a variety of services and features. The main categories include the following:

- **Analytics** enables the embedding of analytics in applications.
- **DevOps** simplifies application development and operations.
- **Integration** allows integrating local and cloud applications.
- **Mobile**, enables the development of mobile applications.
- **User experience** enables personalized and simple user interactions.

> Although SAP Cloud Platform has a similar name to SAP HANA Enterprise Cloud (HEC), the two platforms have different intentions and purposes.
>
> Both are variants of HANA cloud technology but the two products use different service models. While SCP provides a PaaS tool for developing and running cloud-based applications, HEC is an Infrastructure-as-a-Service (IaaS) tool that enables companies to run SAP-based operations in a hosted environment. SAP hosts HEC applications in multiple data centers around the world and provides ongoing application support and management, including upgrades, backups, patches, recovery, infrastructure monitoring, and event detection.

SAP Cloud Platform can also be integrated with other applications and can be used by non-SAP customers. SAP Cloud Platform can be deployed on any of the three major public cloud infrastructure providers: Amazon Web Services (AWS), Microsoft Azure, and Google.

Data Services from SAP

Table 3.11 shows an overview of the data services of the SAP BTP.

SAP Data Hub and SAP Data Intelligence

SAP Data Hub (now renamed “SAP Data Intelligence” in the cloud-only version) is a new platform that virtually integrates data from enterprise applications, such as SAP S/4HANA, and distributed systems such as Hadoop via pipelines and a governance support model, and has since expanded into a machine learning platform. During the writing of this book alone, the orientation of the Data Hub changed from a tool for “Unified Data Integration for SAP” to an “AI and Information Management Platform” (see Fig. 3.6 for an overview).

Table 3.11 Overview of SAP BTP data services

Big Data Services	Fully managed Hadoop and Spark systems
MongoDB Service	NoSQL database
Object Store Service	Supports storage and management of unstructured data (files, BLOBs)
PostgreSQL Service	PostgreSQL database
Redis Service	Implementation of an in-memory cache layer with Redis
SAP ASE	SAP ASE database (in-house development)
SAP HANA Service	In-memory database fully managed across multiple clouds
SAP HANA Spatial Services	Business-enabled spatial data services
SAP Data Warehouse cloud	Data Warehouse as Hybrid Cloud Service
SAP Data Hub	Comprehensive data integration as a service

SAP SCP offers a competitive range of data services

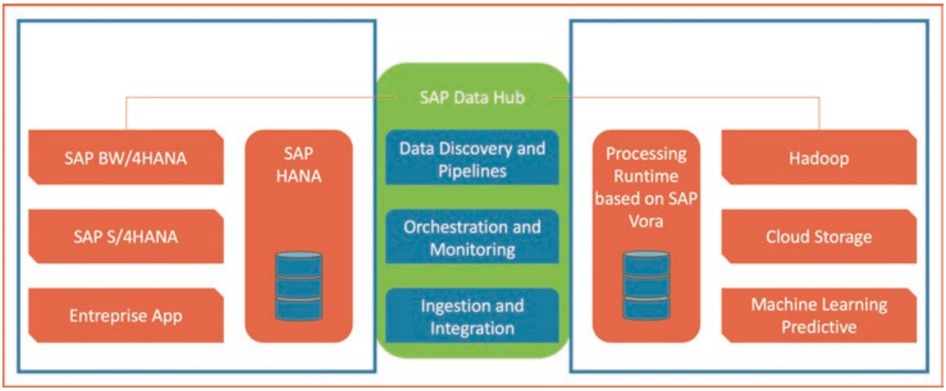

SAP Data Hub
SAP BW/4HANA
SAP S/4HANA
Entreprise App
SAP HANA
Data Discovery and Pipelines
Orchestration and Monitoring
Ingestion and Integration
Processing Runtime based on SAP Vora
Hadoop
Cloud Storage
Machine Learning Predictive

Fig. 3.6 Overview of the integration with the SAP Data Hub

An example architecture where enterprise data is integrated with a distributed system (originating from an IoT network or elsewhere) on the SAP Data Hub is shown in Fig. 3.6. In the past, data was often replicated to the applications that needed it to run with some sort of ETL (extract, transform, load) tool. This is no longer possible with big data, as the challenge is already to store it and make it accessible once, e.g., via MapReduce in Apache Hadoop. Therefore, this data is accessed on-premise via SAP Data Hub. SAP Data Hub is based on the SAP HANA in-memory database and is integrated with SAP Vora, a data management and integration platform. On the front end, it uses a simple desktop design environment or cockpit. The cockpit allows users to create data pipelines, view all connected systems and the status of connections, and display the underlying data systems of the source. Drag-and-drop features allow users to create graphical data flow models. Although SAP Data Hub manages data from a variety of sources, the data itself is never removed from the native source. This "push-down" model means that processing is distributed to the native data source, resulting in faster processing and return of results. It also means that users may be able to take advantage of serverless cloud computing to reduce data management costs.

The biggest potential benefit of SAP Data Hub is that it allows organizations to manage and leverage the vast amounts of data that reside in their various enterprise systems, data lakes, and data warehouses. Without a way to manage the flow of this data, it may not be very useful but SAP Data Hub allows companies to build applications that can gain insights from the data. Data governance in SAP Data Hub also allows companies to ensure that the data lineage is genuine, that it is secure, and that it provides privacy protection. The software also shows who has access to the data, who is using it, who has changed it, where it came from, and where it is going. SAP Data Hub can run on-premises or in the cloud and hybrid environments and supports the following services [10]:

- **Data pipelines** across data lakes (based on Hadoop), object stores (Amazon S3), cloud/on-premise databases, and data warehouses. From the outset, the solution spans the entire data landscape and leverages the "push-down" feature of distributed computing:
  - Execution of data transformations, data quality, and data preparation processes via a graphical user interface.
  - Definition of data pipelines and streams.
  - Embed and produce data scientist scripts, programs, and algorithms.
  - Creation of open libraries or ML algorithms in a framework.
- **Orchestration of** complex processes and workflows across system boundaries
  - Workflow creation of workflows and processes across the landscape with monitoring and analysis capabilities.
  - Execution of end-to-end data processes, starting with the ingestion of data into the landscape (e.g. the data lakes) including data processing, up to the.
  - Provision or integration of the resulting data into business processes and applications.
  - Remote Process Planning: Process chains in the SAP Business Warehouse, data flows with SAP Data Services and flowgraphs with SAP HANA Smart Data Integration.
- **Data ingestion and processing** for data lakes, support of unstructured and structured data/files or streams
  - Pre-built functions for data integration, cleansing, enrichment, masking, and anonymization.
  - No coding or scripting to prepare and transform data in data lakes.
  - Kafka stream integration in end-to-end data pipelines.
  - Data quality and data governance functions are executed in the data lake in a distributed manner through integrated services that are extensible through open source components or cloud micro-services.
  - Use and integration of SAP HANA Smart Data Integration, SAP Data Services, SAP Business Warehouse (BW).
- **Machine learning and data manipulation development**

  - Automated machine learning - Use machine learning methods and algorithms without data science expertise using preconfigured functions.
  - Workflow automation - enables automation of workflows from data preparation to model selection, validation, and deployment.
  - Support for open technologies and developer tools - a wide range of open-source and SAP technologies are available in the platform, including SAP HANA, Python, Apache Spark, and TensorFlow.
  - Easy data preparation and exploration - allows data to be selected and prepared with an intuitive user interface, without the need for technical requirements or programming skills
  - Bring-your-own-model - deploying and reusing existing machine learning models, orchestrating data flows to and from them and integrating with surrounding systems with APIs.
  - Data science tools, so JupyterLab is available natively.
- **Integration and provision of services and interfaces**
  - The created models can be made available directly to surrounding systems via an interface.
  - Support for SAP's own (ODATA, SAP Cloud Data Integration) and open standards (JSON, RESTful, XML).
  - Native integration with SAP Cloud (access to SAP Cloud systems and services).
- **Control, management, operationalization, and creation of** complex data landscapes
  - Handling connections between systems with a supplied adapter for connections.
  - Unified landscape monitoring and planning provides a single point of entry where data stewards can view the status of data processes across all connected components.
  - Predefined adapter framework for connectivity.
  - Set up and manage zones in a landscape (e.g., lab environment, production, etc.) with associated policies and service levels.
  - Security and access control features.
- **Metadata lifecycle** with lineage and impact analysis
  - Metadata model content creation with repository integration (based on GitHub).
- **Data discovery** to visually understand the value in data lake data.
  - Data profiles for large data sets with quality and comprehensive structural information.
  - Ability to crawl, recognize and mark data elements.
  - Disclosure of the found data for further use.

ML Services from SAP

The core aspects of the ML services offered by SAP are distributed between the on-premise solutions on the SAP HANA database platform section "Overview of Other Microsoft Azure Services" and the services in the SAP HANA Cloud Platform. Other SAP HANA components not listed in Table 3.12, such as the Automated Predictive Library (APL) delivered with SAP HANA, can also be used to build machine learning models. Integrating

Table 3.12 Overview of the ML services of the SCP

SAP Conversational AI	AI chatbots using an end-to-end collaborative platform.
SAP Leonardo ML Foundation	The SAP Leonardo Machine Learning Foundation, available on the SAP Cloud Platform, contains a library of prebuilt machine learning models that can be accessed via REST APIs. The platform also allows you to deploy your models
SAP Predictive Service	Web services are used to perform predictive analytics on data resources stored in the SAP HANA database in the cloud.
SAP HANA Predictive Analytics Library (PAL)	Part of the SAP HANA database section "Overview of Other Microsoft Azure Services" and the Application Function Library (AFL) contained therein, which contains functions that can be called within SAP HANA using SQL scripts to execute machine learning algorithms.

SCP also structures ML service offerings into four categories. PAL is special because it is included in the SAP HANA database platform

SAP HANA applications with the R programming language allows machine learning algorithms and the associated model to be integrated into SAP. The technology behind AFL allows you to build external libraries of machine learning algorithms that can be linked to SAP HANA applications.

SAP Leonardo Machine Learning Foundation

The SAP Leonardo Machine Learning Foundation is a machine learning and data science platform that enables users, developers, customers, and partners to build intelligent applications. SAP Leonardo Machine Learning Foundation supports multiple functions for developers and data scientists - from using customizable functional services for text, image, and language processing to training and deploying their deep learning models for use in enterprise applications such as the SAP ERP system S/4HANA. The SAP Leonardo ML Foundation consists of "functional services" and "core capabilities". These services and capabilities are provided as an API for developers to build intelligent processes by embedding them into their applications. The "Functional Service APIs" are divided into three broad groups, namely: image-related, language-related, and text-related. Machine learning models have been developed for each service offering and their services can be aggregated via REST APIs.

In addition, data scientists need the capabilities to train, adapt existing models, or create their models. These are available via the "core capabilities" of the ML Foundation. Typically, an SAP S/4HANA system connects to ML Foundation via the cloud platform connector or SAP Data Hub. The available ML Foundation services and capabilities help customers test and deploy solutions within a short period. SAP itself divides the services into three categories: SAP Leonardo Machine Learning Functional Services, SAP Leonardo Machine Learning Predictive Services, and SAP Leonardo Machine Learning Business Services - which will be presented below:

SAP Leonardo Machine Learning Functional Services - These out-of-the-box machine learning models can be accessed via REST APIs. REST technology is preferred because it uses less bandwidth and is, therefore, more suitable for internet use. APIs are available for image classification, image feature extraction, document feature extraction, and much more. The application of image recognition has far-reaching implications. It can be used for brand recognition and track a company's visibility on social media, such as data on Facebook or Twitter. Machine learning models can be retrained using company-specific data. Customers and partners can also use their machine learning models in a well-defined governance process. Some of the functional services available include the following applications:

- Image classification: calculates and returns classifications with their probabilities for a given image.
- Image feature extraction: ability to extract feature vectors for any image that can be used for comparison, information retrieval, clustering, or further processing.
- Topic recognition: extracts topics from documents and rates them according to the most relevant topics.

SAP Leonardo Machine Learning Predictive Services - Analytics services, such as classification, clustering, and outlier detection can be performed on the SAP HANA database on SAP Cloud Platform. On the SAP Cloud Platform, various models can be created and integrated from structured data to generate classifications, scores, and recommendations. SAP Leonardo Machine Learning Predictive Services have been available in SAP HANA for some time. However, SAP Leonardo has increased the availability of algorithms to run predictive services. SAP Predictive Analytics Integrator Service is part of SAP Leonardo Machine Learning Predictive Services and integrates and enables predictive models for cloud applications. The framework enables enterprises to deploy predictive models in enterprise applications. Enterprise users can use the results of the models in the form of recommendations and decisions to help the business strategize and improve.

SAP Leonardo Machine Learning Business Services - These business application services are available to provide better insight into the financial health of business processes, customers, and suppliers. Currently, SAP Leonardo Machine Learning Business Services offers three APIs that are available for use on SAP APIs:

- SAP Intelligent Financing API: This API provides custom models and recommendation systems in specific functions and industries. Some examples of SAP Leonardo Machine Learning intelligent applications are resume matching and invoice and cash matching. SAP Intelligent Financing can also calculate a replacement credit score by analyzing its business activities across a business network (suppliers and buyers). This calculation can result in an index called an SAP Finance Health Score, which represents the sustainability of a business entity. One advantage of an SAP Finance Health Score is that a supplier with a high score can obtain a lower interest rate and a longer credit period when seeking financing from banks.

- SAP Service Ticket Intelligence Classification API: An interface for improving customer service with machine learning support.
- SAP Service Ticket Intelligence API: This recommendation API makes recommendations based on a machine learning model to improve customer service for a business.

AP Predictive Service

The SAP BTP Predictive Service is a group of Web services[21] that perform predictive analytics on data resources stored in the SAP HANA database in the cloud. These services are provided as REST APIs that can be used to deploy and use the predictive analytics capabilities in the cloud application.

The Predictive Service handles the complexity of developing services directly on the Predictive Model Engine. Web services are easy to understand and provide a simple programming paradigm adapted for web applications in the cloud. The predictive service supports CRUD (create, read, update, delete) operations with data over HTTPS, sending requests and receiving responses only in JSON format. The Predictive Service must be deployed as an application on the SAP Cloud Platform instance before it can be used. After deployment, a schema is created in the database to store the data used by the service such as service call history and order results. The data resources are records stored in SAP HANA database tables. One must be registered in the database schema, which may be different from that of the service. Each time the service is called, a predictive model is created from the dataset using SAP HANA APL. The results returned by the service are retrieved from the predictive model. The predictive service provides synchronous and asynchronous predictive model execution. For models that require a long processing time, asynchronous execution provides a better user experience. A service returns some kind of insight. Each service runs the data mining process. Each service allows you to refine the creation of the model, for example, by using the automatic variable selection feature, by correcting variable descriptions, or by ignoring columns that contain unnecessary information, such as column identifiers. SAP BTP Cloud Platform predictive services can integrate the following capabilities into any application on SAP BTP:

- **Time series forecasting**: This allows you to analyze time series and forecast future values based on the trend, periodicities, and fluctuations found in previous data. As an example, this API allows you to predict how many products to order based on forecasted demand. To do this, it is possible to identify important influencing factors by having the API analyze within the dataset which variables most influence the target/output set and how.

[21] See service description at: https://api.sap.com/api/Predictive_Services/resource

- **Outlier detection**: identifies records in the aggregate whose destination is significantly different from what is expected. This allows potentially fraudulent financial transactions based on anomalous activity to be detected.
- **Model-based "what-if" simulations**: includes simulation of changes to determine which variables and objects will be affected and how. This service answers the question: what are the critical variables of a data set and its values if some of the values of a variable change? This makes it easy to investigate and validate hidden relationships between planned actions and the resulting consequences.
- **Application of predictive models**: This API enables a model to be trained based on historical customer data to be able to make a prediction. This model can now be accessed via an API and provides predictions for a future event based on the historical and trained data.

SAP HANA Database Platform

In this context, **SAP HANA** is the lower-level column-oriented and relational in-memory database management system for all applications [11].

Its main function is as a database server for storing and retrieving data from the deployed applications (such as the SAP ERP system S/4HANA). In addition, it is also designed to perform advanced analytics (predictive analytics, geospatial data processing, text analytics, text search, streaming analytics, or graphics data processing) and includes its application server. Thus, SAP HANA as a platform, both as a fast data source and as an application server, is able to equip a server-based JavaScript application server with the SAP HANA XS JavaScript engine, including a front-end solution that can be implemented with SAPUI5, a proprietary HTML5 and JavaScript framework [12, 13].

Interestingly, a separate BI or BA application does not seem to be envisaged in the reference architecture. Rather, it is the case that the required analyses are distributed across the various applications. For example, analyses of sales data are possible in the SAP Customer Activity Repository, analyses of customer data in SAP Hybris Commerce, and analysis and customer segmentation in the context of marketing activities in SAP Hybris Marketing.

With HANA 2.0 SP2, it is possible to call TensorFlow models. HANA, thus, provides a method for invoking external machine learning (EML) models via a remote source. The EML integration is done via a wrapper function that is very similar to the predictive analysis library (PAL) or business function library (BFL). Like PAL and BFL, EML is table-based, with tables storing model metadata, parameters, input data, and output results. At the lowest level, EML models are created and accessed via SQL, making it a perfect building block.

Build or Buy?

From a vendor perspective, the aforementioned managed ML services are positioned for organizations that are in the process of building their data science teams or whose teams are primarily comprised of data analysts, BI specialists, or software engineers (who may be transitioning to data science). However, even small to mid-sized data science teams can gain value from evaluating these machine learning as a service (MLaaS) and data storage as a service (DSaaS) offerings. Because these providers can produce and access so much data, they can build and train their machine learning models in-house - these pre-built models can quickly provide higher performance. It also makes sense to use these MLaaS offerings as a basis for comparison with in-house models.

For example, let us say that the BA team is tasked with automating the labeling of fashion products in the online store. This essentially means that a product, such as a dress is used as input to automatically and accurately determine attributes, such as sleeve length (without), neckline (scoop neck), length (mini dress), pattern (stripes), color (green, yellow, red), etc. These attributes are then used to either personalize the marketing campaign or improve the search function.

Using machine learning - specifically computer vision - to handle this product attribute classification process makes sense because of the (assumed) large, diverse product catalog. Access to these attributes helps the company with some important initiatives:

- Personalization around content and product recommendations
- Improving discoverability and search in the user experience
- Forecasting/planning for inventory management

The following presents a framework particularly for evaluating various MLaaS products. One will notice how the attributes for transparency, ease of use, flexibility, cost, and performance span across each part of the workflow. The evaluation framework and its questions help understand whether using an MLaaS product is the right approach for the task and the team - from the data that is injected into the MLaaS offering to the actual results of the analytics applications. To prevent these systems from becoming expensive black boxes, one should also ask how the evaluation is actually done and whether the results can be updated and used in an automated way in further executing systems (ERP, CRM, etc.).

The evaluation framework is based on the business analytics model for artificial intelligence (BAM.AI) procedure model presented above and is divided into the two main topics "Development" (section "Development Cycle") and "Deployment" (section "Deployment Cycle") but extends these to include the decision perspective of data management in section "Data Management". This covers all relevant aspects. For all three main topics, the points of cost, performance (capability), transparency and traceability, usability, and flexibility must be evaluated (see Table 3.13).

Table 3.13 Evaluation framework under BAM.AI

	Data management	Development	Deployment
Costs	What are the costs associated with the data transfer? What are the costs of storing the data? Are there costs to integrate systems (development effort or licenses)?	What is the pricing model for training and evaluating the models? What is the licensing model for users/developers?	What is the pricing model for predictions?
Performance (capability)	Is (automatic) data cleansing or data completion supported? How much data is needed to train the models?	How will the models generated be assessed/evaluated? How long do the training phases last?	Are the models retrained in real-time or periodically? Does a change (training) lead to unavailability? How is the transition regulated? What do the SLAs look like?
Transparency	Which functions, frameworks, or algorithms are supported? What requirements? Own models and algorithms?	Which algorithms and frameworks are supported? Are the results transparent and comprehensible?	Is there monitoring? Is there any reporting? What does a possible escalation (downtime, errors, etc.) look like?
Usability	Is there support for data cleansing, data completion, or data preprocessing?	Can the model be tested? Can an error be detected and corrected quickly? Also externally? Which and how many variables can be manipulated and how?	Can results (models and data) also be migrated from or to other providers?
Flexibility	Which data formats are supported? Do suitable APIs exist for importing and later exporting data? What are the restrictions?	How often are the models renewed? Can this be regulated? Can the models be re-trained? Is there a possibility to define exceptions?	Is integration with other tools and software solutions possible? Are there APIs?

Five criteria can be used to support the selection of a platform provider

References

1. Dixon, J.: Pentaho, Hadoop, and Data Lakes. https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/ (2010). Accessed on 1 Mar 2016
2. Cafarella, M., Lorica, B., Cutting, D.: The Next 10 Years of Apache Hadoop. https://www.oreilly.com/ideas/the-next-10-years-of-apache-hadoop (2016). Accessed on 18 Mar 2019
3. Ruder, S.: Highlights of NIPS 2016: Adversarial Learning, Meta-Learning, and More. http://ruder.io/highlights-nips-2016/index.html#thenutsandboltsofmachinelearning (2016). Accessed on 23 May 2019
4. Pattanayak, S., John, S.: Pro Deep Learning with TensorFlow. Springer, Berlin (2017)
5. Google: Ocado: Delivering Big Results by Learning from Big Data. https://cloud.google.com/customers/ocado/ (2018). Accessed on 12 Feb 2019
6. Sato, K.: Using Machine Learning for Insurance Pricing Optimization. https://cloud.google.com/blog/products/gcp/using-machine-learning-for-insurance-pricing-optimization (2017). Accessed on 1 Feb 2019
7. Google Inc.: Introducing Google App Engine + Our New Blog. http://googleappengine.blogspot.com/2008/04/introducing-google-app-engine-our-new.html (2008). Accessed on 22 Jan 2018
8. Brewer, E.: Towards robust towards robust distributed systems 19th ACM symposium on principles of distributed computing (PODC). Invited talk (2000)
9. Brewer, E.: CAP twelve years later: how the. Computer. **2**, 23–29 (2012)
10. Hartz, M.: What Is SAP Data Hub? And Answers to Other Frequently Asked Questions – SAP HANA (2017)
11. Plattner, H., Leukert, B.: The in-Memory Revolution: how SAP HANA Enables Business of the Future. Springer, Berlin (2015)
12. Prassol, P.: In-memory-platform SAP HANA als big data-Anwendungsplattform. In: Fasel, D., Meier, A. (eds.) Big Data: Grundlagen, Systeme und Nutzungspotenziale, pp. 195–209. Springer, Wiesbaden (2016)
13. Prassol, P.: SAP HANA als Anwendungsplattform für Real-Time Business. HMD Praxis der Wirtschaftsinformatik. **52**(3), 358–372 (2015)

Case Studies on the Use of AI-Based Business Analytics

4

Case Study: Analyzing Customer Sentiment in Real Time with Streaming Analytics

Customer Satisfaction in the Retail Sector

The changes in the retail sector in recent years have been manifold: globalization has led to international competitors gaining access to markets and now being able to offer products of the same quality at lower prices. In addition, other market players, especially from e-commerce, are entering the already highly competitive market, thus intensifying competition. This and other factors have led to increasing predatory competition, not only due to overcapacity but also due to stagnating overall market volumes. The scope for price differentiation has become smaller overall, as “technical-functional harmonization” has taken place in many product categories. Due to the similarity of product ranges and types of business in the retail sector - especially in the food sector - the retail price is largely the only remaining competitive instrument. The increased price transparency of consumers is based on the increased transparency due to the ubiquitous availability of the internet and price search engines and comparisons. The (positive) correlation between customer satisfaction and willingness to buy is scientifically almost undisputed [1–5] and could be used as another competitive tool in the German food retail sector besides price orientation. The changed customer behavior and the broader availability of information should not only benefit the customers but also serve the retailers to their advantage. The presented research project presents a case study of stationary retail with an existing retail system architecture based on standard software.

There are various studies on customer satisfaction in retail and their most important influencing factors. Although a generalization across all retail industries and distribution

F. Weber, *Artificial Intelligence for Business Analytics*,
https://doi.org/10.1007/978-3-658-37599-7_4

types is not permissible, the performance criteria are weighted differently but always include the following factors: price, availability, and advice [6, 7].

The discrepancy between reality, real customer satisfaction, and perceived reality mean that the assumptions made by retailers about customer satisfaction cannot deviate further. For example, a study by Germany's largest retail association [8] found that the gap between consumer demand and actual fulfillment by retailers is large for many product categories. In many cases, as a look at young consumers in the "smart natives" category shows, it will become even larger in the future if retailers do not act. The low level of attention and deviation in brick-and-mortar retailing is due in part to the fact that most methods known to date for measuring and evaluating customer satisfaction are based on direct customer surveys [9], which are operationally difficult and costly to conduct in brick-and-mortar retailing. Today, customer satisfaction in brick-and-mortar retailing is mainly collected for benchmarking and company comparisons and determined by external service providers on an individual or aggregated level. In addition, the employees in the branches are currently the only contacts who receive customer feedback and can react to it. However, the employees usually do not have the necessary time - at least this is not explicitly regulated by most retailers - to actively receive and pass on customer feedback. In most cases, the response to a customer's feedback depends on the situation but a structured and analytical process is lost in the daily workflow and effort.

In contrast, Amazon, as a ubiquitous example of online retail and a large emerging competitor for brick-and-mortar retail, has a strong customer focus, and customer feedback is collected and analyzed at every possible point of customer interaction [10]. In addition to the approach of collecting customer feedback and sentiment data, the retail giant also relies on a large knowledge base and the ability to analyze this data. As the implementation of omnichannel scenarios blurs the boundaries between online and offline retail, this could be a major disadvantage for traditional retailers.

Technology Acceptance and Omnichannel for More Data

The technological change relevant for the retail industry is mainly driven by the widespread adoption of the internet and new technologies, such as smart mobile devices (handhelds, smartphones, and tablets) and corresponding software applications (mobile apps, mobile payments, e-marketing, or location-based services). This increasing adoption of mobile devices and ubiquitous access to the internet not only increases transparency in terms of prices and information for customers but also enables retail companies to connect with their customers. In addition to allowing brick-and-mortar retailers to individually identify their customers, these technological advances can be used as a data source to collect customer-related data. Social media, mobile apps, and the Internet of Things (IoT) enable the collection of relevant data.

In recent years, the number of participants and thus the intensity of use of **social media** has increased significantly. More than 500 million tweets (6000 per second) are created daily. This mass customization is an opportunity for marketing and consumer research. There is now an entire field of research called “social media analytics.” In addition to data volume and variety, velocity, known as the three “Vs,” characterizes the term “big data.” In recent years, big data analytics, which has mostly worked through offline data processing, has expanded to include batch processing, online processing, and streaming. In general, controlling social media networks is impossible and therefore the best strategy is to use the networks for one’s benefit [11]. There are two main strategies: active use as a marketing tool or passive use as a data source. This data can be used for a variety of purposes: alerting, text analysis, sentiment analysis, social network analysis, trend analysis, attribute analysis, association analysis, and text mining. When these use cases of big data analytics are executed in real time, they are referred to as streaming analytics.

With the advent of **omnichannel retailing**, **mobile devices** play a key role in all available channels [12]. Not only personal mobile devices, smartphones, and tablets are relevant sources of customer data but also handheld shopping assistants and self-scanners or self-checkout devices. Combined personal and non-personal mobile devices cover all areas of human interaction [13] in retail. There are many possible use cases: shopping list applications, shopping assistants, navigation, payments, or self-checkout. All these possible use cases allow the integration of capturing active and passive customer sentiment and feedback data.

Another way to collect the data needed is through the use and integration of **IoT** and hardware at the **“Point Of Sale (POS)**. Hardware-based feedback systems include modified POS hardware that allows each customer to provide feedback immediately at the checkout. Specialized hardware with the one-way case of collecting feedback (see Fig. 4.1) can also be deployed.

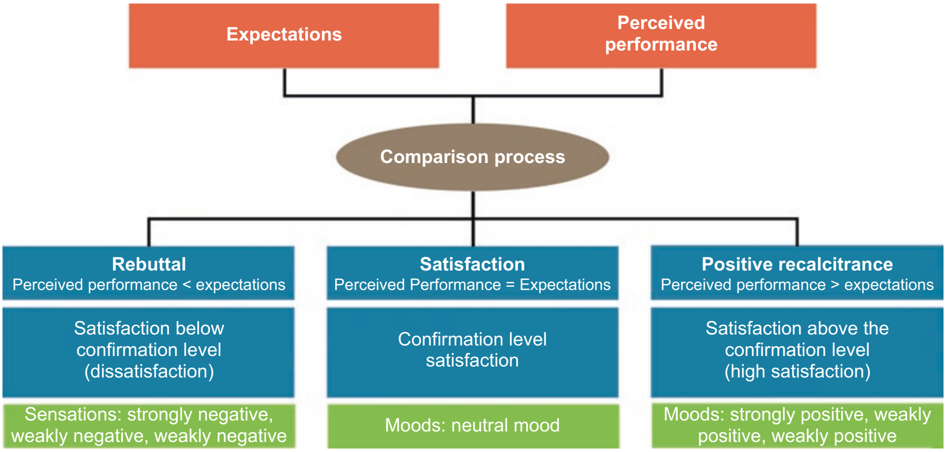

Expectations
Perceived performance
Comparison process
Rebuttal
Perceived performance < expectations
Satisfaction
Perceived Performance = Expectations
Positive recalcitrance
Perceived performance > expectations
Satisfaction below confirmation level (dissatisfaction)
Confirmation level satisfaction
Satisfaction above the confirmation level (high satisfaction)
Sensations: strongly negative, weakly negative, weakly negative
Moods: neutral mood
Moods: strongly positive, weakly positive, weakly positive

Fig. 4.1 Schematic representation of the confirmation/de confirmation paradigm

Customer Satisfaction Streaming Index (CSSI)

In empirical marketing and behavioral research, a variety of different definitions and models of customer satisfaction have developed over time. However, there is an abstract agreement that customer satisfaction is not a directly observable, hypothetical construct. A simple explanatory approach in terms of the basic model that can be derived without burdening deeper theoretical and psychological backgrounds at this point is the “confirmation/disconfirmation paradigm” [14–16]. In this paradigm, customer satisfaction is the result of a psychological construct based on a comparison between the actual experience with a service or product (perceived performance) and a certain standard of comparison (expectations) (a schematic overview can be found in Fig. 4.1). If the comparison of the perceived performance corresponds to the expected target performance, confirmation occurs. Depending on the operationalization of this construct, an evaluation at the level of a specific situation, industry, assortment, product, or the entire shopping experience is made possible. Based on this scientifically accepted paradigm [17], a retail customer satisfaction assessment can be developed using data from social media, IoT sources, and mobile devices.

There are already various methods and procedures used in retail to determine customer satisfaction, store evaluation, or price image [18] but they are operationalized questionnaires or direct customer surveys that cannot be mapped in real-time. Since no specific questions can be asked but the data depends on the active input of the customers, there is neither standardization nor operationalization in the sense of the classical question instruments. Therefore, only a basic tendency can be derived and the developed index is based on the well-known approach of the customer satisfaction index. Analytically all indices work after the same basic structure: from a set (n) of evaluation characteristics (k) for a set (m) of questioned customers (i) the weighted ($Wk_{,i}$) average value of the single evaluations (Z) is formed. To match the characteristics of the source medium, the importance levels of the individual evaluation scores ($Wk_{,i}$) are weighted over time using the fitted exponential function of time difference. Timeliness is a crucial variable in the short-lived social media environment, which is why older posts have a lower weighting.

This procedure makes it possible to evaluate a current result with maximum importance (e.g. a branch that needs cleaning up) but to adjust it over time due to its decreasing importance (branch has already been cleaned up but the negative impression remains). Normalization is performed for a few variables analogous to the variables used in the sentiment analysis (“strongly positive” to “strongly negative”). The CSSI (see Fig. 4.2) contains a geographical delimitation and is not surveyed comprehensively but by branch and region.

$$\text{CSSI} = \sum\nolimits_{k-m} \sum\nolimits_{i=1-n} (Z_{k,i} * (W_{k,i} * \frac{1}{e^{t_n - t_{n-1}}}))$$

n = Number of evaluation characteristics
m = Number of customers
Z = Satisfaction value per evaluation point and customer
W = Importance per evaluation point and customer
k = Evaluation characteristic
i = Customer
t_n = Evaluation date
t_{n-1} = Time of entry of the evaluation

Fig. 4.2 Customer Satisfaction Streaming Index (CSSI)

Implementation in a Retail Architecture

Some events, and thus their data representation, can be characterized as an "endless" stream of parallel or sequential instances. To perform batch processing on this type of data, the data must be stored, the collection stopped, and the data processed. This process is then restarted for the next batch run. In contrast, stream processing of data streams as an unbroken chain of events in near real-time. Over the years, numerous streaming platforms have been developed that follow the streaming paradigm. This is based on the goal of querying a continuous stream of data and detecting conditions within a short time. These platforms are all suitable for the above objective of determining the customer sentiment index. However, the information obtained in this way is of limited use when considered individually. Much more crucial than the pure metric itself is the cause-effect relationship leading to it. For this reason, the cause-effect analysis in the BI environment is usually the downstream and still completely manual follow-up process for all activities. Only with a possible domain- and application-specific cause-effect relationship is any kind of real-time analysis useful. In the case described, a decreasing CSSI without possible causes cannot be used for a reaction (see Fig. 4.3).

Therefore, integration into the existing system architecture in retail is required. In order to map the above-mentioned decisive influencing factors of customer satisfaction in retail, product availability, and price, integration into the existing processes and systems of merchandise management is required. The system architectures that are still technologically and architecturally designed for batch processes [19] today are a challenge that should not be underestimated due to the enormous volume, e.g. of POS data in food retailing at peak times with several checkouts in several thousand stores.

The implementation in system architecture (see Fig. 4.4) must meet the existing requirements and conditions of the retail industry. The standard software provider SAP cites "The SAP Model Company for Omnichannel Retail" [20] as a reference architecture for retail

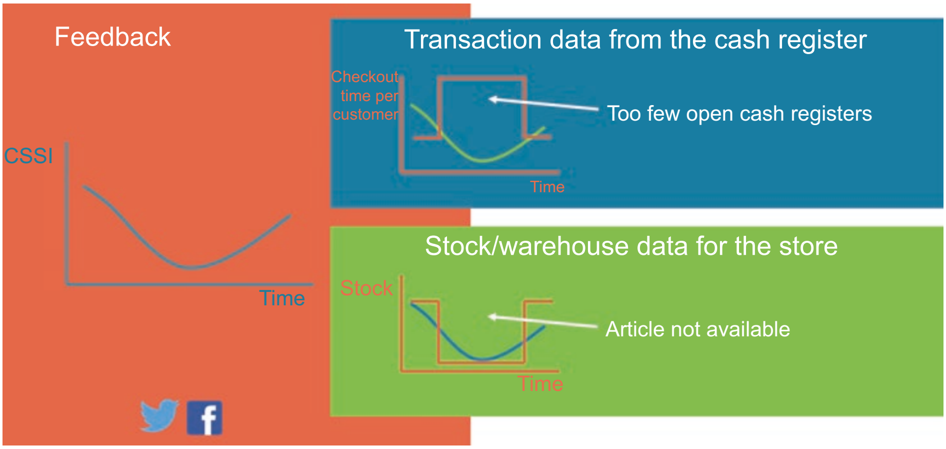

Feedback
CSSI
Time
Transaction data from the cash register
Checkout time per customer
Too few open cash registers
Time
Stock/warehouse data for the store
Stock
Article not available
Time

Fig. 4.3 CSSI in use in stationary retail trade

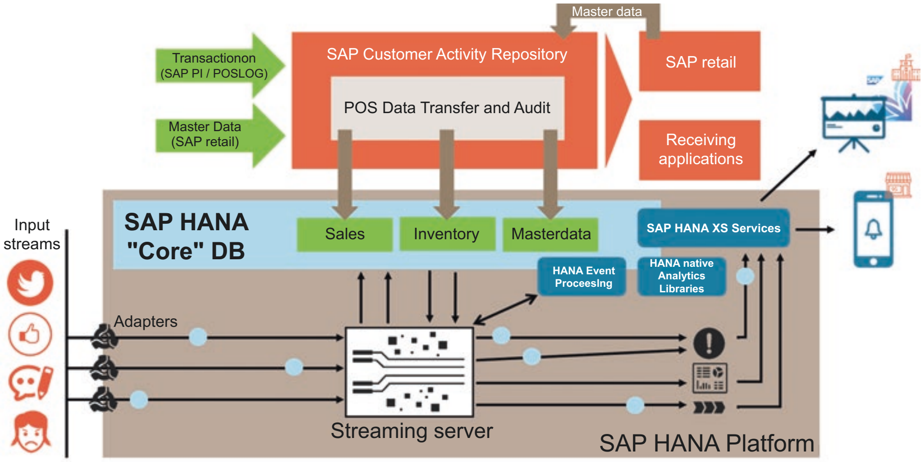


Fig. 4.4 System architecture based on the retail reference architecture

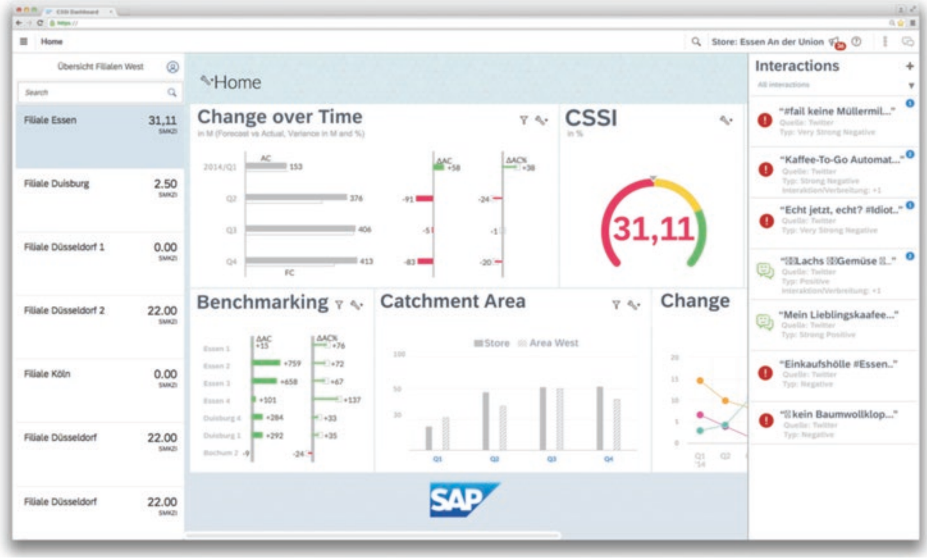
companies. The SAP Customer Activity Repository provides harmonized data models and functionality for all multichannel transactional data in real-time for internal use or consuming applications. SAP HANA is the lower-level in-memory database management system [21] used for all applications. It primarily serves as a database server but is also designed for advanced analytics and includes an application server. The technical implementation is done in the mentioned system architecture. On a technical level, streaming analysis was implemented with the use and customization of "smart data integration" on the SAP Streaming Server integrated into the platform. In a first step, the incoming social media posts, tweets, are sorted by appropriate keywords and geographic assignment to a store. The tweets are then subjected to sentiment analysis, which provides the results of the named sentiments, events, organizations, and situations. In case of a crucial event (very negative sentiment, wrong article, the importance of the sender, etc.), this information is sent in real-time to the appropriate recipients via a push notification. All events are also included in the calculation of the industry-specific CSSI index. A more in-depth analysis is possibly analogous to a drill-down. In this way, each event is enriched with corresponding detailed information (out-of-stock, price changes, or duration of the payout processes at the time of the event) from the SAP CAR (Customer Activity Repository).

Results

This case study is about one of the largest German retailers. The German retail sector is characterized by an oligopolistic market with strong intracompetition between existing retailers and increasing competition between traditional and new “pure” digital players [22]. With the withdrawal of Kaiser’s Tengelmann, one of the last large independent retail chains, in 2016 and the upcoming market entry of Amazon with Amazon Fresh, this competition has further intensified. Furthermore, decreasing possibilities for differentiation between types of operation [23], increased costs, an overall increase in price awareness [24], and the influence of the company’s price image on the customer’s choice for a retail chain can be observed.

The retail group studied here operates several different sales lines, hypermarkets, supermarkets, and discount formats with large assortment widths and depths: 8000 products in the discount chain, around 35,000 products in the normal supermarket chain, and up to 80,000 products in the hypermarket chain. In addition, each store serves up to 10,000 customers daily. An initial test of the above prototype implementation was conducted, analyzing approximately 250,000 tweets over 11 months in 2017. Sentiment analysis within the SAP HANA database and streaming service examined 70% of the messages that could be rated. Overall, 32% of all messages were rated as “strongly positive,” 20% were rated as “strongly negative,” 12% were rated as “weakly negative,” 3% were rated as “small problem,” another 3% were rated as “big problem,” and the rest were rated as “weakly negative” or “weakly positive.” The last set of messages received a rating of “neutral.” Based on this data, the CSSI was calculated and provided to business customers. The resulting customer satisfaction information can be accessed. This can be done via a dashboard (Fig. 4.5) or a direct push notification to the end-user. Depending on the user group, there are different possible uses. In the store itself, the alert function is the central interaction point for the store management. However, access to the dashboard can also be used at any time to receive regular evaluations or to obtain more detailed information in the event of an alert. At a higher organizational level, it is possible to compare the different branches with each other.

The prototype implementation of the newly formulated customer satisfaction streaming index (CSSI) is a general analytical approach for dealing with customer feedback. The presented approach is a suitable way to aggregate the inputs generated by customers on



Filiale Essen	31,11 SMKZI
Filiale Duisburg	2.50 SMKZI
Filiale Düsseldorf 1	0.00 SMKZI
Filiale Düsseldorf 2	22.00 SMKZI
Filiale Köln	0.00 SMKZI
Filiale Düsseldorf	22.00 SMKZI
Filiale Düsseldorf	22.00 SMKZI

"#fail keine Müllermil..."
Quelle: Twitter
Typ: Very Strong Negative

"Kaffee-To-Go Automat..."
Quelle: Twitter
Typ: Strong Negative
Interaktion/Verbreitung: +1

"Echt jetzt, echt? #Idiot..."
Quelle: Twitter
Typ: Very Strong Negative

"Lachs Gemüse ..."
Quelle: Twitter
Typ: Positive
Interaktion/Verbreitung: +1

"Mein Lieblingskaafee..."
Quelle: Twitter
Typ: Strong Positive

"Einkaufshölle #Essen..."
Quelle: Twitter
Typ: Negative

"kein Baumwollklop..."
Quelle: Twitter
Typ: Negative

Fig. 4.5 Dashboard shows the Customer Satisfaction Streaming Index (CSSI)

various social media sources. The CSSI, as a general benchmark for customer assessment, works very well and can provide an opportunity to at least partially bridge the information gap between online and offline retailers. The collected data can also be seen as a step to address the current low adoption of artificial intelligence and machine learning in the retail sector [25]. Integration with machine learning models could be a suitable use case to improve marketing atomization, for example. The idea of direct notification of sales staff is partly difficult to handle. First, the necessary technical equipment is not available. Second, distribution of mobile devices to retail sales staff has traditionally been low or non-existent at the case study company. In reviewing the initial data processed, it seems likely that content and frequency tweets would enable real-time monitoring for individual store locations but the overall sentiment is very well represented. However, due to the low usage of tweets with a location tag in Germany, selecting a unique store is challenging. Text analysis based on machine learning could help here but it is not clear if this process is feasible. Another drawback is the fact that Facebook has restricted access to a variety of search APIs after the so-called "Cambridge Analytica scandal". Facebook's significantly higher market share over Twitter in the German market poses a major challenge to data available for this project. This leaves the use of in-store IoT devices as the only way to collect customer feedback. In a second research step, an empirical study needs to be conducted. A simple button at the checkout would be the easiest way to collect the relevant data.

With the presented adaptation of the known metrics for measuring customer satisfaction to the CSSI and the integration of the existing information from the transaction and master data systems in retail, the retailer only can use the CSSI as a KPI for operational control. It enables real-time responses to events and can be used to show underlying trends over time. Benchmarking between different trades is also possible. Initial practical tests have shown that KPI calculations work well but an empirical test of the proposed real-time alerting needs to be conducted to uncover several previously undiscovered issues. How the CSSI developed here performs against a proven method in marketing and behavioral research has yet to be tested with empirical studies. What is certain, however, is that CSSI and the above implementations offer a unique opportunity for brick-and-mortar retailers to respond to customer feedback in real-time.

Case Study: Market Segmentation and Automation in Retailing with Neural Networks

Technology has finally advanced to the point where the largest data sets can be processed for marketing in a way that makes sense to the customer and is profitable for the company. Even though more than 90 percent of retail executives believe that personalization in marketing is their top priority, only a handful deliver on this aspect. For brick-and-mortar retailers, in particular, a personalized approach to their customers is troublesome. Due to historically low demand and the unique situation of physical store retailers, customers are largely unknown and unidentifiable to retail companies. Only the use of customer/loyalty cards or coupons offers a possibility for customer identification but even this simple method is largely unused in German food retailing. Marketing measures are not tailored to individual stores and the corresponding catchment area but a large-scale network based on an arbitrarily defined distribution area. This is the result of various circumstances but most importantly, the required data is largely unavailable and no processes have yet been established to automate marketing management. The low use of methods from advanced analytics and artificial intelligence [25] is also part of a larger knowledge deficit in these areas of retail. Capturing external data, particularly from the internet, could be a way to develop a more personalized understanding to enable marketing to brick-and-mortar customers. Although the acquisition of potentially valuable data is rather straightforward, the resulting data quality and nature pose a problem for many clustering algorithms. Even machine learning methods that are considered more robust to noise have problems with the unpredictability of such data acquisition. For example, the mapping capability of artificial neural networks (ANNs) depends on their predefined structure (i.e., the number of layers or the number of hidden units); for an unpredictable and rapidly changing data collection, this requires a lot of adaptation.

In this case study, we present a method to collect a large amount of relevant location information for a large German brick-and-mortar retailer from online data sources, including internal location data (such as assortment, sales history, price history), competition, social structure (from income to housing situation), and demographics. This dataset is mapped to a network model of Germany at the 1 km^2 level. Based on this granular data, we create a clustering model of similar stores of the retail chain. The clustering is performed using machine learning methods, in particular, growing neural gas (GNG), a neural network variant. The learning algorithm of GNG is also called incremental learning, which characteristically changes the network topology in the learning process. This leads to certain independence from assumptions made in advance about the data structure and relationship.

The resulting store clusters are then tested for location-specific marketing. This separation of marketing activities from nationwide marketing campaigns to store clusters is seen as improving optimization (e.g., an A-B test can be performed between stores of the same class), and eventually full automation can be performed. Finally, the results are discussed considering the current practice and experience of the responsible local marketing managers.

The Location Decision in Stationary Trade

The choice of a suitable store location is one of the most important decisions in stationary retail. The term "location" is used in the retail literature in two ways. On the one hand, it is the geographical location of a retail company where it concentrates its production factors and wants to establish contact with consumers. On the other hand, it is about all the decisions and issues surrounding the placement of central warehouses. This article deals only with the sales locations in the stationary retail trade (also shops, commercial buildings, shopping centers, etc.). The location of retail businesses whose operations are sales-oriented and whose sales areas are all characterized by relatively narrow boundaries must be clearly determined by the primacy of sales. Location decisions have a lasting impact on sales results. The characteristic of the location influencing the sales result is thus undisputed [26] so that the retail location - subject to the possibility of free choice - must be regarded as an instrument of sales policy. Its importance in comparison with the other instruments of distribution policy is all the greater because its duration and intensity are much greater than those of any other instrument. For example, free and advertising policy measures, assortment policy, and service policy have more or less narrow lateral limits. They can be continuously adapted to market needs so that even a measure that does not correspond to the market can in most cases be revised and changed without consequences that threaten the existence of the firm. The constructive character of a location policy decision and the amount of investment involved, on the other hand, prohibit a short-term review, since misalignment of the location policy can simultaneously lead to the closure of the company. Within the marketing mix, location policy plays a dominant role because it modifies the other sub-policies. Especially the assortment and pricing policy is closely linked to the location quality. For example, a high-quality city location requires a different assortment level and assortment dimension than a peripheral location or a location in a suburb; the proximity of a department store also determines the price level and assortment of a specialty store.

In this case study, it is assumed that the retailer's choice of location cannot be influenced and has already been implemented. Thus, the setting describes a situation where the stores exist and are already in operation. Furthermore, the current marketing process uses the set locations as the basis for local marketing activities. The store locations are used to define the catchment area, and this catchment area is the segmentation basis for all marketing activities. A further segmentation, the narrowing of the catchment area into different sub-segments, is currently not part of most marketing processes in stationary retail.

Marketing Segmentation and Catchment Area

The Catchment Area of the Bricks-and-Mortar Trade Marketing decisions are limited by the catchment area of one's location and also those of companies that may be competitors. This is necessary because the competitors relevant to competition also cover the overlapping catchment area geographically and the market potential must therefore be calculated differently. The catchment area of a location includes all the locations (homes or offices) of all potential customers. Several methods are available for determining the catchment area. These are systematized differently in the literature. In general, there are two different models: macro and micro market areas [27, 28]. A macro market area is the sales area of an entire city, a historically developed retail settlement, or a shopping center. The delineation of the market area at this level of aggregation takes into account the overall attractiveness of the retail agglomeration rather than company-specific factors. A micro market area is the distribution area of a single retail company. Company-specific characteristics such as the type of company are included in the analysis. In principle, the requirements for the delineation of a macro market area are likely to be lower than for a micro-market area. This is because micro-market areas require demand to be known, whereas the definition of the macro-market area includes only the demands attributable to the entire area, regardless of how demand is distributed among competing stores and businesses.

Marketing and Market Segmentation The market segmentation concept was first developed in the 1950s for consumer goods marketing and has been an integral part of marketing science for many years [29]. In the specific design of individual business relationships today, it is economically essential to evaluate relationships with customers and customer groups in order to allocate scarce resources, such as marketing and sales budgets and sales visit times (which in turn can be expressed in monetary units) to their greatest possible use. Information on the optimal design of marketing and sales policies is important for business practice, not least because the optimal allocation of scarce resources to customers offers significantly greater potential for profit increases than determining the optimal total amount [30]. Market segmentation can be viewed as the process of dividing a large market into smaller groups or clusters [31, 32].

The level of detail of segmentation can range from classic segment marketing to niche marketing to individual marketing. The recommended approach for segmenting markets is often described in the literature with the model "Segmenting, Targeting, Positioning" (STP) [33]. This approach divides the process of market segmentation into three main steps that must be performed in chronological order. The first step is the actual segmentation, i.e. the division of the overall market into individual segments by using appropriate segmentation variables. These segments optimally represent buyer groups that are as clearly distinguishable as possible, each of which is to be addressed with a range of services or a marketing mix that is specially tailored to them. To assess the opportunities in each submarket, a supplier must now evaluate the attractiveness of the segments and, based on this evaluation, determine which segments to serve. The third step is to develop a positioning concept for each target market to establish a sustainable, competitive position that will be signaled to the targeted consumers. Although in some cases segmentation in service markets and retail markets are also treated separately, there are hardly any specific segmentation criteria or approaches for the latter two areas. Rather, researchers assume that the concepts developed for B2C physical goods markets can also be used for consumer services and retail [33]. Several advantages can be derived from market segmentation. The most important advantage is that decision-makers can target a smaller market with higher precision. In this way, resources can be used more wisely and efficiently. In addition, market segmentation leads to closer relationships between customers and the company. Moreover, the results of market segmentation can be used for decision-makers to determine the respective competitive strategies (i.e., differentiation, low-cost, or focus st**rategy).**

Classical Clustering Approaches and Growing Neural Gas

Clustering analysis is a technique used to divide a set of objects into k groups so that each group is uniform with respect to certain attributes based on the specific criteria. Cluster analysis aims to make it a popular marketing segmentation tool. Clustering algorithms can be generally classified as partitioning methods (e.g., K-means), hierarchical methods (e.g., agglomerative approach), density-based methods (e.g., Gaussian mixture models), and grid-based methods (e.g., self-organizing maps - SOMs) [34, 35].

Heuristic approaches - The most commonly used method for defining groups of branch clusters is the manual heuristic approach. Based on a few descriptive categories, the clusters are delineated from each other. Most often, the resulting clusters follow a practical pattern related to the organizational environment of the company. For example, the number of groups follows geographical or regional aspects. Other groupings may follow the logic of different store types or differences within assortments or store layouts.

K-means - The basic idea of agglomerative methods is the successive combination of the most similar observation units [36]. After units have been combined into aggregates, they are not re-sorted into different aggregates during the agglomeration process but are combined as a whole into larger aggregates in subsequent steps. This results in the hierarchical system of aggregates described above, which forms a rough sequence of partitions of the sample. K-means, on the other hand, causes a given partition to be optimized by a series of repositioning steps of individual observations from one aggregate to another. The number of aggregates remains unchanged. The optimality criterion is a measure of aggregate heterogeneity and partitions, namely the sum of squares. This measure is a so-called index. The smaller this index is, the more homogeneous the aggregates are and the better they can be interpreted as clusters. One, therefore, looks for the partition with the smallest index, given by the number of aggregates.

Kohonen Networks or Self-Organizing Maps (SOM). Kohonen networks are a type of unsupervised learning algorithm that reveals structures in data. They are also referred to as self-organizing maps (SOM) [37]. The input vectors are compared with the weight vectors. Usually, the Euclidean distance is the measure of similarity. The neuron with the smallest distance or the highest similarity to the input pattern wins and receives all the activation. The weight of the winning neuron in the input layer is modified to further increase the similarity. Geometrically, the algorithm shifts the weight vector towards the input vector. So far, the Kohonen network works like other clustering methods. To generate the topological structure, the weight vectors of the neighboring neurons of the winner also change. This requires a definition of quarters that implement various functions, such as the Gaussian function or the cosine. These functions provide a measure of the distance of each neuron in a layer from the winner neuron, which affects the intensity of the weight change. The closer a neuron is to the activated neuron, the more its weight vector is adjusted. The vectors of very close neurons are therefore always shifted in similar directions. This creates clusters in which similar patterns are mapped [37].

Problems with the classical clustering approaches. In the context of data analysis, problems are reduced to a set of the most informative features that are expected to best describe the objects. This type of abstraction is necessary to use these smaller sets of features to form a vector in a multidimensional space of features. Then, some similarity between these vectors is measured. This similarity can be measured quantitatively by the proximity (distance) of the vectors to each other in space. The proximity between these vectors within this space is then used to determine the similarity; this is usually understood as the Euclidean distance or the Mahalanobis distance if there is a correlation between the vectors. With big data, the amount of input data is now much larger as more information and data can be captured and used. The attributes to describe an object or problem now become much more available. However, Mitsyn and Ososkov [38] see this availability creating further difficulties in solving this type of problem:

- The number of measurements to be processed becomes extremely large (up to 106 and more).
- New data sources and measurements will become dynamically available over time.
- The space used for grouping input data into regions has many more dimensions.
- No preliminary information is available on the number and location of the regions sought.

This is where neural gas algorithms come into play. Unlike Kohonen networks or SOM, neural gas is intended for data clustering rather than data visualization, so it can use an optimal data grid and is not limited by topological constraints. Therefore, in terms of quantization error or classification, better results can often be obtained [39]. The ability of machine-associated methods to learn and recognize peculiar classes of objects in a complex and noisy space of parameters, and to learn the hidden relationships between parameters of objects, has been shown to be particularly suitable for the problem of neural networks and neural gas classification [40].

Neural gas and growing neural gas. Neural gas (NG) is based on works by Martinetz and Schulten [41]. It is a single-layer neural network trained with an unsupervised learning process. The weights of the neurons correspond to the vectors in the input space. Each neuron represents a subspace of the input space in which all data points have a smaller Euclidean distance to their weight vector than to the weight vectors of the other neurons. The growing neural gas (GNG) is an incremental neural network that can be used with either a supervised or unsupervised learning procedure, depending on the problem. It was developed in the early 1990s by Fritzke [42]. The number of layers of adaptable weights depends on the training algorithm used. Supervised learning requires two layers since its basic structure is strongly based on RBF networks. In contrast, unsupervised training requires only one layer analogous to carbon maps and neural gas. The goal of an unsupervised GNG is to represent an input data distribution with minimal quantization error; in contrast, a supervised GNG should classify input data with as little error as possible. The neurons of the GNG layer each represent a subspace of the input space in which all data points have a smaller Euclidean distance to their weight vector than to the weight vectors of other neurons. That is, a Voronoi decomposition of the input space takes place. Moreover, the neurons of the GNG layer correspond to nodes in an undirected graph whose edges are constructed depending on the neighborhood of the weight vectors in the input space. The incremental architecture of the growing neural gas allows an optimal adaptation of the network topology and the structure of the graph to the problem at hand. For this purpose, starting from a minimal number of consecutive neurons can be inserted into the GNG layer but can also be implicitly removed. The criteria for controlling network growth depend on the basic paradigm (supervised or unsupervised). Accumulator variables and neighborhood relations in the input space are crucial for the growth process. Neurons are deployed where the accumulated error is largest. This reduces the total error of the network. The structure of the graph is continuously adjusted. All edges have an aging parameter. Edges that have reached a maximum age are deleted. Nodes (or neurons) that no longer have edges are also removed. An explicit mechanism for deleting nodes is not required.

Growing neural gases for clustering. The extension and application of GNG after the initial idea [41] are diverse. GNG is also used for a variety of application scenarios, such as soft competitive learning [43]. All recent developments in GNN, including supervised and unsupervised learning, are studied in Xinjian, Guojian [44]. The suitability of GNG for clustering is shown in a variety of different domains, even for astronomical images [45] or clustering of the data stream [46].

Project Structure

In light of the work outlined above, the following research questions are proposed:

- (**Objective 1) Is it possible to collect enough data from the web to build store clusters?** It is not clear whether it is possible to rely on freely available data from the web to build store clusters for the case study company. Large market research companies have business models based on the fact that retail companies currently rely solely on their data to make such decisions. We want to investigate whether this dependency is justified or whether more independence and thus potential cost savings and competitive advantage are possible.
- (**Objective 2**) **Can the neural gas algorithm generate suitable memory clusters that have high intraclass and low interclass similarity?** As mentioned earlier, many algorithms and methods for clustering already exist in the literature. Our resulting clusters need to be tested against standard cluster quality measures to assess the utility of the proposed method. The ability to adapt to changing data inputs and be robust to noise is not enough to justify the use of the algorithm in practice.
- **(Objective 3) Are the resulting store clusters suitable for segmenting marketing activities and thus for use as a basis for marketing automation in stationary retail?** The resulting clusters must be tested not only for mathematical-theoretical evaluation but also for practical applicability and usefulness within current marketing practice.

The Data and Sources

Several authors [47, 48] described the availability of relevant information in price management, an integral part of marketing mix decisions, as a basic requirement for effective design. Basically, a distinction can be made between internal information (about products, customers, costs, stores) and external information (about the market structure, the competitive environment or the target group and customer behavior) as well as primary statistical and secondary statistical information [49], whereby the integration of external and internal information represents one of the greatest challenges in marketing [50]. Primary statistics is the form of (statistical) data collection that is specifically and exclusively for statistical purposes, such as a census. Secondary statistics is a form of statistical collection that essentially consists of the transmission of data that was not originally collected for statistical purposes, e.g. the transmission of billing data for statistical purposes (Table 4.1).

The use of primary statistical information sources usually has the advantage that this data is already available since it was originally collected for a purpose other than pricing. On the other hand, the collection of secondary statistical information requires separate processes to be set up or purchased as an external service.

Directly available data comes from multiple sources that provide processed and cleaned data that is readily available and usually identifiable by geo-coordinates or name tags

Table 4.1 Overview of available data sources and the information they contain

	Internal ERP system	Master data
Directly available (primary statistics)	Census Bureau 2011	1 km network, inhabitants, foreigners, gender, age, household size, vacancy rate, living space, average living space
	Federal Statistical Office	Average household income
	Local registries	Average household income
	Internal ERP system	Master data
Web scraping (secondary statistics)	Websites	Master data, event data, competition data
	Twitter API	Evaluations, sensations
	Facebook API	Evaluations, sensations
	Google places API	Reviews, contests
	Real estate platforms (partly API)	Average rent, property quality

Overview of the data sources used. A distinction can be made here between primary and secondary statistical information.

(cities or other geographic markers). The standard source in current retail marketing practice is to obtain data from large marketing and market research firms, such as The Nielsen Company. By default, data is only available at the 5-digit zip code area and municipality level. The internal ERP system already provides the locations and rudimentary information about the operational stores. This includes name, location, type, sales data, and assortment. The Bureau of the Census in Germany provides a collection of information gathered from the most recent census in Germany, which took place in 2011. Each resident is assigned to an address and thus to a grid cell with a side length of 1 km. Members of the armed forces, police, and foreign service working abroad and their dependents are not taken into account in this evaluation. Foreigners are persons with foreign citizenship. In the classification of nationality, a distinction is made between persons with German and foreign nationality. Persons with German nationality are considered Germans, regardless of the existence of other nationalities. The sex of each person was recorded as ‘male’ or ‘female’ in the 2011 census. No further specifications are planned, as this corresponds to the information provided by the population registration offices on May 9, 2011. The average age of the population (in years) is the ratio of the sum of the age in years of the total population and the total population per grid cell. The average household size is the ratio of the number of all persons living in private households to the total number of private households per km^2 and is based on data from the 2011 census. Second homes are also taken into account in this context. A private household consists of at least one person. This is based on the “concept of shared living”. All persons living together in the same dwelling, regardless of their residential status (main/secondary residence), are considered members of the same private household, so that there is one private household per occupied dwelling. Persons in collective and institutional accommodation are not included here but only persons with a self-managed household. The vacancy rate (dwellings) is the ratio of vacant dwellings to all occupied and vacant dwellings per grid cell, expressed as a percentage. It does not include the following dwellings: vacation and recreational dwellings, diplomatic/foreign armed forces dwellings, and commercial dwellings. The calculation is made for dwellings in residential buildings (excluding hostels) and is based on data from the last census in Germany (2011). The living area is the floor area of the entire apartment in m^2. The apartment also includes rooms outside the actual enclosure (e.g. attics) as well as basement and floor rooms developed for residential purposes. For the purpose of determining the usable floor area, the rooms are counted as follows:

- Full: the floor areas of rooms/parts of rooms with a clear height of at least 2 meters;
- Half: the floor areas of rooms/parts of rooms with a clear height of at least 1 meter but less than 2 meters; unheatable conservatories, swimming pools, and similar rooms closed on all sides; normally a quarter but not more than half: the areas of balconies, loggias, roof gardens, and terraces.

Average living space per inhabitant is the ratio of the total living space of occupied dwellings in m^2 to the total number of persons in occupied dwellings per grid cell and is based on data from the 2011 census, excluding diplomatic/foreign armed forces housing, holiday and leisure housing, and commercial housing. The calculation is made for dwellings in residential buildings (excluding hostels). The average or median income in a society or group is the income level from which the number of low-income households (or individuals) is equal to that of higher-income households. The ratings of mostly anonymous web users are collected via the APIs Twitter, Google Places, and Facebook. Here, a numerical rating and a textual overview of each place are available. It is important to note that these reviews are pre-selected and censored, as all major platforms use automatic spam detection methods to delete reviews that are likely to be spam. Based on the collected review data, the sentiments are calculated and stored on an individual and aggregated level for each place. From various event and news platforms, such as Eventbrite, Eventim, or local newspaper announcements, information about local events with their dates, locations, and categories are scraped and matched to the grid map.

The same process is also provided for the competitive data. This includes local shops, restaurants, and other types of business. In this process, all the information available on the various platforms is combined into a single data set. The datasets are then matched against the list of stores within geographic distance. From a number of different German real estate platforms, such as immowelt or immobilienscout24, the average rental price and the property quality are determined for each grid map. Here, the street names within the grid are used to calculate an overall mean value for these two key figures.

The grid. All these data were collected and then incorporated into the core concept of a raster-based map of Germany at the granularity level of 1 km^2 (see Fig. 4.6 for a visualization).

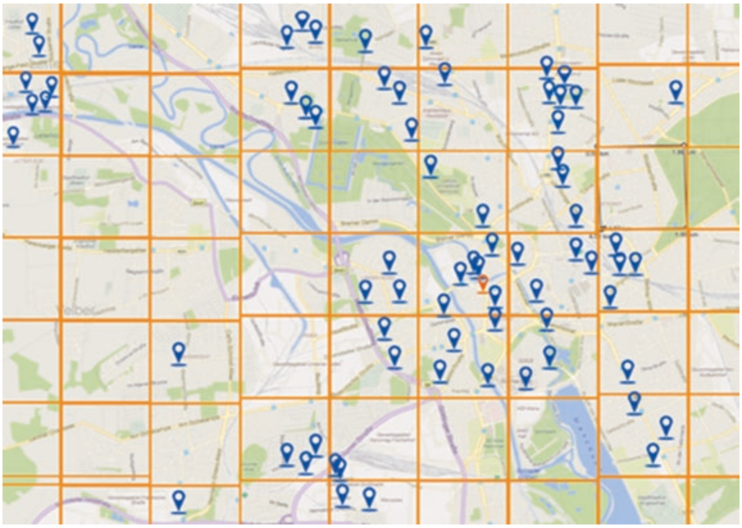


Fig. 4.6 Visualization of the base map (1 km fields)

The image above shows the visualization of a random sample area. Each of the fields in this grid contains the following data: grid ID, geographic location (latitude and longitude), population, number of foreigners, gender percentage, average age, household size, vacancy rate, living area, average living area, average household income, average rent, and property quality. The stores, competitors, and events are then grouped by their location into a single grid ID.

Implementation

In order to collect data from a wide variety of websites, several scraping bots were implemented (the overall architecture is outlined in Fig. 4.7). Web services are the de facto standard for integrating data and services. However, there are data integration scenarios that web services cannot fully address. Some internet databases and tools do not support web services, and existing web services do not meet user data requirements. As a result, web data scraping, one of the oldest web content extraction techniques, is still able to provide a valuable service for a variety of applications ranging from simple extraction automata to online meta-servers. Since much of the data needed in this project was not directly available, simply because there is no API, most of the data were collected through scraping bots (independent computer program to collect data from a specific website or a set of similar websites). Here, a single bot is set up using the Python programming language for each data source that automatically searches and extracts the required data. In the next step, the data is cleaned and validated. Then, the data is exported to the central

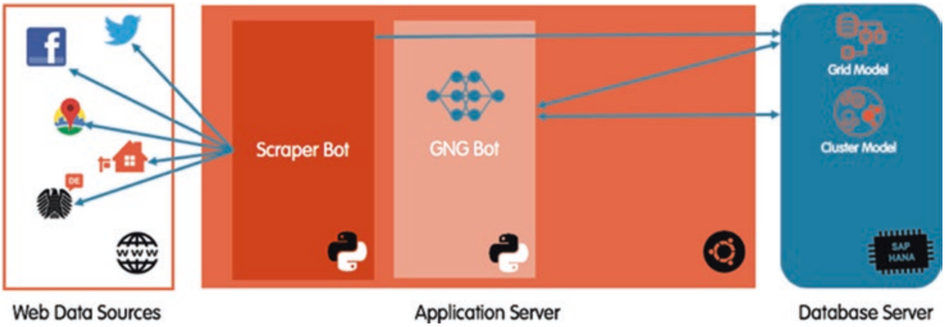


Fig. 4.7 System architecture of the implementation

SAP HANA in-memory database. The same procedure is set for the data sources for which a publicly available API is available. All bots are built to be checked regularly for possible updates and data changes. With the native streaming server, this process could also be set up in real-time as events (data) come into the system [51]. Due to the nature of the relatively stable dataset used here, we decided against this idea for the first prototype.

Based on the available data from the central database, the bot for the growing neural gas is determined. In this phase, the newly available data is added to the existing network and the cluster is re-evaluated.

The central database serves not only as a data repository but also for sentiment analysis of the collected review data sets.

Growing gas neural algorithm and approach. According to [42], the GNG algorithm is represented by the following pseudocode:

1. One starts with two units a and b at random positions w_a and w_b in Rn. The matrices must also store connection information and connection age information. In addition, each neuron has an error counter representing the cumulative error.
2. One generates an input vector ξ according to P(ξ).
3. One searches for the closest unit s1 and the second closest unit s2.
4. Now, the age of all connections emitting s1 is increased. Also, summarize the error counter with Δerror(s1) = ||ws1 -ξ||||2.
5. We now calculate the displacement of s1 and its direct topological neighbors in the direction ξ with $_{\Delta ws1}$ = εb(ξ - ws1); $_{\Delta wn}$ = εn(ξ - $_{wn}$), where εb εn ∈ (0, 1) is a parameter to set the motion.
6. The age of the connections is reset afterward if s1 and s2 were connected, otherwise, you create a connection. All connections with age higher than the predefined maximum age $_{amax}$ and also neurons that have no outgoing connections are deleted. The age of all outgoing connections of s1 is increased by one.
7. Each λ-iteration locates the neuron q with the largest value of its error counter and its topological neighbor with the highest error counter f. A new neuron r between q and f

with $_{wr}$ = ((wq + wf)/2) is created. Likewise, the connections between r, q, and f and the original connection between q and f is deleted.

The error counter of q and f is reduced by multiplication with α. And the error counter of the new neuron is initialized with the value of f.

8. The error count of all neurons by multiplying by δ is reduced and restarted with step one.

Results

Regarding the first question of the objective, “Is it possible to collect enough data from the web to form store clusters” (**objective 1**), a comparison with related literature is useful. The majority of the literature on new retail store location decisions can make a clear distinction between internal and external (trade area) variables. Looking at the present dataset, it is clear that internal data are disregarded. However, the external dataset is more extensive than much of the existing literature [52, 53]. The less considered internal data can be added from internal resources in an extension of the existing environment. This is where the advantages of GNG algorithms come into play, as they allow smooth adaptation to the extended vectors.

To answer the second objective question, “Can the neural gas algorithm produce suitable clusters that have high intraclass and low interclass similarity?” (**objective 2**), a mathematical test is performed. The resulting clusters are visualized in Fig. 4.8.

A total of nine different clusters out of 3761 stores seems small at first glance but for retail marketing practice, this is quite feasible, as most changes within the marketing mix also require physical changes in the store (change of assortment or placement of products).

Thus, in addition to the mathematical quality specified by the GNG, usability for the stationary retail sector is also possible.

To answer the current question, “Are the resulting store clusters suitable to segment marketing activities and thus to be used as a basis for marketing automation in brick-and-mortar retail?” (objective 3), we follow a guideline developed by Kesting and Rennhak [33].

Criteria for segmentation must meet certain conditions. The literature [54, 55] generally sets six requirements for them, which aim, among other things, to ensure a meaningful market division (see Table 4.2).

As the majority of the selected variables are related to economic (micro and macro) factors, they are linked to the future purchasing behavior of customers within the segments. Since most of the factors come from research institutes or the official German census office, the requirement is met that the factors must be measurable and recordable with existing market research methods. The segments to be targeted are both accessibility and reachability. It is undeniable that targeting requires more effort to be implemented. Since an empirical test is yet to be implemented with the use of the segments in practice,

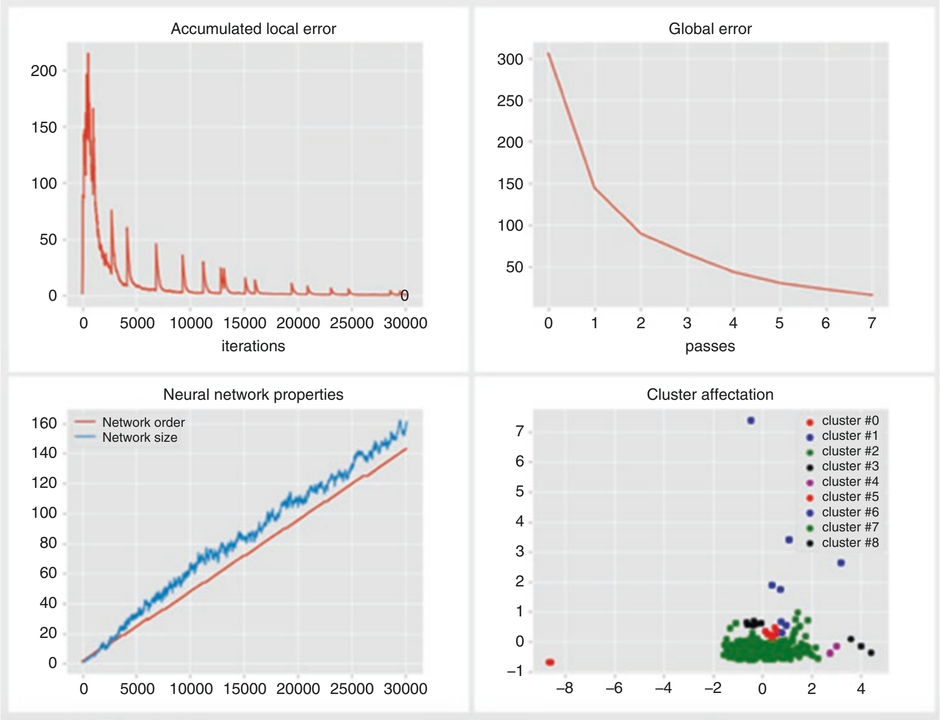


Fig. 4.8 Two-dimensional representation of the data set and the resulting clusters: the cumulative and global error, the order and size of the network, and a higher-level visualization of the resulting clusters

Table 4.2 Segmentation requirements

Relevance for purchasing behavior	Suitable indicators for future purchasing behavior
Measurability (operationality)	Measurable and documentable with existing market research methods
Accessibility	Ensuring a targeted approach to the segments formed
Capacity to act	Ensuring the targeted use of marketing instruments
Economic efficiency	The benefits of the survey should be greater than the associated costs
Temporal stability	Long-term validity of the information collected based on the criteria

Five categories can be used to evaluate segmentation solutions in retail marketing practice.

the assessment of cost-effectiveness is yet to be completed. Also, empirical testing and testing over a longer period are subject to further investigation.

This case study presented an innovative prototype for the use of artificial intelligence in a field that is not known for the widespread use of such technologies [56]. The application of GNG to a core retail marketing task is unique. In particular, the results show that the use of this algorithm can provide useful results in a noisy and dynamic environment. Based on the resulting clusters, a marketing automation approach is possible. Since these clusters can be dynamically adapted to changing external and internal changes, marketing activities can be adjusted on the fly. As a basis, we suggest starting with simple A/B tests to change parameters within the marketing mix (price, product, promotion, or location) and derive tailored marketing activities.

References

1. Hallowell, R.: The relationships of customer satisfaction, customer loyalty, and profitability: an empirical study. Int. J. Serv. Ind. Manag. **7**(4), 27–42 (1996)
2. Homburg, C., Koschate, N., Hoyer, W.D.: Do satisfied customers really pay more? A study of the relationship between customer satisfaction and willingness to pay. J. Mark. **69**(2), 84–96 (2005)
3. Francioni, B., Savelli, E., Cioppi, M.: Store satisfaction and store loyalty: the moderating role of store atmosphere. J. Retail. Consum. Serv. **43**, 333–341 (2018)
4. Kumar, V., Anand, A., Song, H.: Future of retailer profitability: an organizing framework. J. Retail. **93**(1), 96–119 (2017)
5. Anderson, E.W.: Customer satisfaction and price tolerance. Mark. Lett. **7**(3), 265–274 (1996)
6. Renker, C., Maiwald, F.: Vorteilsstrategien des stationären Einzelhandels im Wettbewerb mit dem Online-Handel. In: Binckebanck, L., Elste, R. (Hrsg.) Digitalisierung im Vertrieb: Strategien zum Einsatz neuer Technologien in Vertriebsorganisationen, S. 85–104. Springer, Wiesbaden (2016)
7. Fleer, J.: Kundenzufriedenheit und Kundenloyalität in Multikanalsystemen des Einzelhandels: Eine kaufprozessphasenübergreifende Untersuchung. Springer, Wiesbaden (2016)
8. IFH: Catch me if you can – Wie der stationäre Handel seine Kunden einfangen kann. https://www.cisco.com/c/dam/m/digital/de_emear/1260500/IFH_Kurzstudie_EH_digital_Web.pdf (2017). Zugegriffen am 23.07.2018
9. Töpfer, A.: Konzeptionelle Grundlagen und Messkonzepte für den Kundenzufriedenheitsindex (KZI/CSI) und den Kundenbindungsindex (KBI/CRI). In: Töpfer, A. (Hrsg.) Handbuch Kundenmanagement: Anforderungen, Prozesse, Zufriedenheit, Bindung und Wert von Kunden., S. 309–382. Springer, Berlin/Heidelberg (2008)
10. Anders, G.: Inside Amazon's Idea Machine: How Bezos Decodes Customers. https://www.forbes.com/sites/georgeanders/2012/04/04/inside-amazon/#73807ee56199 (2012). Accessed on 20 May 2018
11. Constantinides, E., Romero, C.L., Boria, M.A.G.: Social media: a new frontier for retailers? Eur. Retail Res. **22**, 1–28 (2008)
12. Piotrowicz, W., Cuthbertson, R.: Introduction to the special issue information technology in retail: toward omnichannel retailing. Int. J. Electron. Commer. **18**(4), 5–16 (2014)
13. Evanschitzky, H., et al.: Consumer trial, continuous use, and economic benefits of a retail service innovation: the case of the personal shopping assistant. J. Prod. Innov. Manag. **32**(3), 459–475 (2015)
14. Oliver, R.L.: Effect of expectation and disconfirmation on postexposure product evaluations: an alternative interpretation. J. Appl. Psychol. **62**(4), 480 (1977)

15. Bösener, K.: Kundenzufriedenheit, Kundenbegeisterung und Kundenpreisverhalten: Empirische Studien zur Untersuchung der Wirkungszusammenhänge. Springer, Berlin (2014)
16. Simon, A., et al.: Safety and usability evaluation of a web-based insulin self-titration system for patients with type 2 diabetes mellitus. Artif. Intell. Med. **59**(1), 23–31 (2013)
17. Fornell, C., et al.: The American customer satisfaction index: nature, purpose, and findings. J. Mark. **60**(4), 7–18 (1996)
18. Becker, J., Schütte, R.: Handelsinformationssysteme. Domänenorientierte Einführung in die Wirtschaftsinformatik, 2. Aufl., Redline-Wirtschaft, Frankfurt am Main (2004)
19. Schütte, R.: Analyse des Einsatzpotenzials von In-Memory-Technologien in Handelsinformationssystemen. In: IMDM (2011)
20. Woesner, I.: Retail Omnichannel Commerce – Model Company. https://www.brainbi.dev (2016). Accessed on 1 July 2017
21. Plattner, H., Leukert, B.: The in-Memory Revolution: how SAP HANA Enables Business of the Future. Springer, Berlin (2015)
22. Schütte, R., Vetter, T.: Analyse des Digitalisierungspotentials von Handelsunternehmen. In: Handel 4.0, S. 75–113. Springer, Berlin (2017)
23. Meffert, H., Burmann, C., Kirchgeorg, M.: Marketing: Grundlagen marktorientierter Unternehmensführung. Konzepte – Instrumente – Praxisbeispiele, 12. Aufl., S. 357–768. Springer Fachmedien, Wiesbaden (2015)
24. Daurer, S., Molitor, D., Spann, M.: Digitalisierung und Konvergenz von Online-und Offline-Welt. Z Betriebswirtsch. **82**(4), 3–23 (2012)
25. Weber, F., Schütte, R.: A domain-oriented analysis of the impact of machine learning – the case of retailing. Big Data Cogn. Comput. **3**(1), 11 (2019)
26. Kari, M., Weber, F., Schütte, R.: Datengetriebene Entscheidungsfindung aus strategischer und operativer Perspektive im Handel. Springer, Berlin (2019). HMD Praxis der Wirtschaftsinformatik
27. Schöler, K.: Das Marktgebiet im Einzelhandel: Determinanten, Erklärungsmodelle u. Gestaltungsmöglichkeiten d. räumlichen Absatzes. Duncker & Humblot, Berlin (1981)
28. Schröder, H.: Handelsmarketing Methoden und Instrumente im Einzelhandel, 1. Aufl. Redline Wirtschaft, München (2002)
29. Wedel, M., Kamakura, W.A.: Market Segmentation: Conceptual and Methodological Foundations, vol. Bd. 8. Springer Science & Business Media, New York (2012)
30. Doyle, P., Saunders, J.: Multiproduct advertising budgeting. Mark. Sci. **9**(2), 97–113 (1990)
31. Smith, W.R.: Product differentiation and market segmentation as alternative marketing strategies. J. Mark. **21**(1), 3–8 (1956)
32. Weinstein, A.: Market Segmentation: Using Niche Marketing to Exploit New Markets. Probus Publishing, Chicago (1987)
33. Kesting, T., Rennhak, C.: Marktsegmentierung in der deutschen Unternehmenspraxis. Springer, Wiesbaden (2008)
34. Huang, J.-J., Tzeng, G.-H., Ong, C.-S.: Marketing segmentation using support vector clustering. Expert Syst. Appl. **32**(2), 313–317 (2007)
35. Jiang, H., Kamber, M.: Data Mining: Concept and Techniques, pp. 26–78. Morgan Kaufmann Publishers Inc., San Francissco (2001)
36. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland (1967)
37. Kohonen, T.: Self-Organization and Associative Memory, vol. Bd. 8. Springer Science & Business Media, New York (2012)
38. Mitsyn, S., Ososkov, G.: The growing neural gas and clustering of large amounts of data. Opt. Mem. Neural Netw. **20**(4), 260–270 (2011)

39. Cottrell, M., et al.: Batch and median neural gas. Neural Netw. **19**(6–7), 762–771 (2006)
40. Brescia, M., et al.: The detection of globular clusters in galaxies as a data mining problem. Mon. Not. R. Astron. Soc. **421**(2), 1155–1165 (2012)
41. Martinetz, T., Schulten, K.: A "Neural-Gas" Network Learns Topologies. MIT Press, Cambridge (1991)
42. Fritzke, B.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems (1995)
43. Chaudhary, V., Ahlawat, A.K., Bhatia, R.: Growing neural networks using soft competitive learning. Int. J. Comput. Appl. (0975–8887). **21**, 1 (2011)
44. Xinjian, Q., Guojian, C., Zheng, W.: An overview of some classical Growing Neural Networks and new developments. In: 2010 2nd International Conference on Education Technology and Computer (2010)
45. Angora, G., et al.: Neural gas based classification of globular clusters. In: International Conference on Data Analytics and Management in Data Intensive Domains. Springer, Berlin (2017)
46. Ghesmoune, M., Lebbah, M., Azzag, H.: A new growing neural gas for clustering data streams. Neural Netw. **78**, 36–50 (2016)
47. Watson, H., Wixom, B.: The current state of business intelligence. Computer. **40**, 96–99 (2007)
48. Awadallah, A., Graham, D.: Hadoop and the Data Warehouse: when to Use which. Copublished by Cloudera, Inc. and Teradata Corporation, California (2011)
49. Hartmann, M.: Preismanagement im Einzelhandel, 1. Aufl., Gabler Edition Wissenschaft (Hrsg.). Dt. Univ.-Verl, Wiesbaden (2006)
50. Weber, F.: Preispolitik. In: Preispolitik im digitalen Zeitalter, pp. 1–12. Springer Gabler, Wiesbaden (2020)
51. Weber, F.: Streaming analytics – real-time customer satisfaction in brick-and-mortar retailing. In: Cybernetics and Automation Control Theory Methods in Intelligent Algorithms. Springer, Cham (2019)
52. Mendes, A.B., Themido, I.H.: Multi-outlet retail site location assessment. Int. Trans. Oper. Res. **11**(1), 1–18 (2004)
53. Themido, I.H., Quintino, A., Leitão, J.: Modelling the retail sales of gasoline in a Portuguese metropolitan area. Int. Trans. Oper. Res. **5**(2), 89–102 (1998)
54. Meffert, H., Burmann, C., Kirchgeorg, M.: Marketing Grundlagen marktorientierter Unternehmensführung, Konzepte, Instrumente, Praxisbeispiele, 9. Aufl. Gabler, Wiesbaden (2000)
55. Freter, H.: Marktsegmentierung (Informationen für Marketing-Entscheidungen). DBW, Stuttgart (1983)
56. Weber, F., Schütte, R.: State-of-the-art and adoption of artificial intelligence in retailing. Digital Policy Regul. Gov. **21**(3), 264–279 (2019)