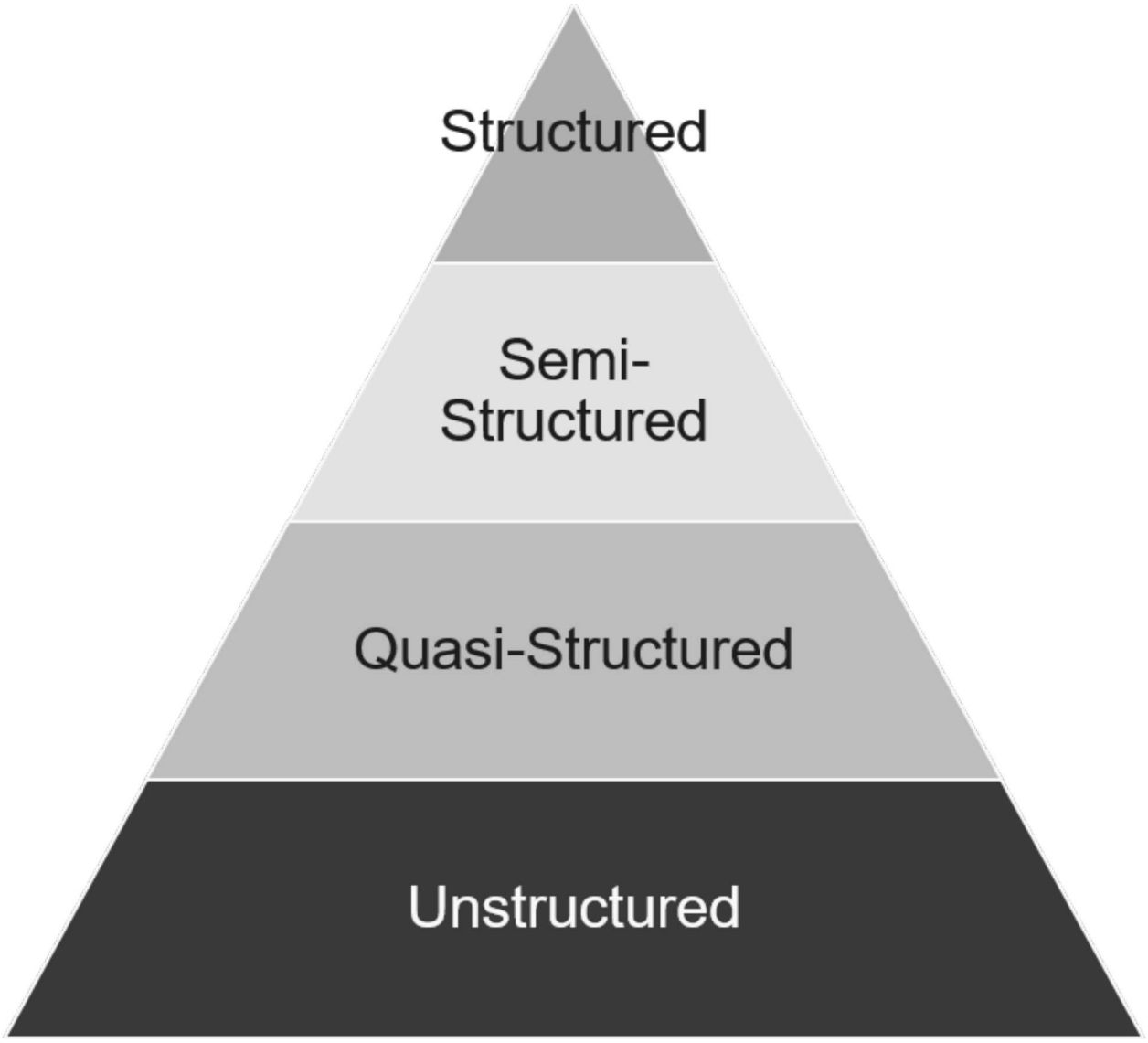
1.1.3 Variety

The variety is the third aspect of big data. Today, the analysis of images, videos and text has become a relatively normal application. However, this was not the case when the term big data was initially coined. Then, most data was held in data warehouses with defined structures, for example, relational databases. The data were mostly numeric or in fixed categories. This began to change when the Internet started to grow into the almost all-encompassing web of knowledge and content we have today. It is no coincidence that Google was at the forefront of the development of big data technologies, because the initial driver was indexing the Internet to enable efficient queries for finding content. This meant that unstructured data, like websites must be indexed and information retrieval algorithms must be executed against this data. The amount of unstructured data is vastly more than that of structured data. Typically, the relation between data structuring and the volume of data is depicted as a pyramid.



At the top of the pyramid is the *structured data*, e.g., tables, comma separated value files and similar. Often, this data can be directly analyzed and pre-processing is only required for data cleaning, e.g., the detection of invalid data or outliers.

Next, we have *semi-structured data*, e.g., XML or JSON files. The main difference between structured and semi-structured data is that semi-structured data formats are often more flexible. For example, each row in the table of a relational database must have values for exactly the same columns. With XML and JSON, the fields are usually

similar, but may have structural differences, e.g., due to optional fields.

The first two layers of the pyramid are defined data formats for which there are usually query languages and/or libraries for the extraction of information. This is not the case on the bottom layers. *Quasi-structured* data has a fixed structure, but not in a convenient and easily accessible data format. For example, consider the output of the `ls -l` command.

```
%ls -l
```

```
total 36
-rw-rw-rw- 1 sherbold sherbold 24996 Mar 26 11:04 01_Introduction.ipynb
-rw-rw-rw- 1 sherbold sherbold  6302 Mar 26 13:32 02_Process.ipynb
drwxrwxrwx 1 sherbold sherbold   512 Mar 26 11:21 images/
```

There certainly is a discernable structure in the output, i.e., *most* lines contain a summary of the user rights, followed by the number of links, the user and group who own the file, the size, the data of the last change, and the name. This structure can be exploited to define a parser for the data, for example, using regular expressions. Thus, we are able to impose a structure on the quasi-structured data, by defining the meaning of the structure on our own and writing our own parsers for the data. A potential problem with quasi-structured data is that these data formats are often not very reliable and may change. For example, `ls` could separate fields with tabulators instead of spaces, which would break most parsers. There is no protection against such changes, which makes such parsers fragile and may mean that significant effort must be invested for the maintenance of parsers for quasi-structured data in production environments.

On the bottom layer of the pyramid is the *unstructured data*, which is the vast majority of available data, e.g., images, videos, and text. The challenge of unstructured data is that a structure must be imposed for analyzing the data. How this is done depends both on the data and the application. Moreover, there are often mixed formats with unstructured data. Just consider this script. We have a mixture of natural language text, images, markdown information that specifies special features of the text (headlines, listing), and even source code. How the structure is imposed depends both on the data, as well as the application.

1.1.4 Innovative forms of information processing

While the three Vs are usually considered as the major aspects of the definition of big data, the other parts of the definition are also important to understand why big data is not just more data, that may be generated rapidly and different formats. The next part of the definition states that *innovative forms of information processing* are required. This means you cannot just use a normal workstation or even a traditional batch system, where you have many computing resources to which a shared storage is attached via the network. Instead, *data locality* becomes an issue, i.e., preventing copies of the data due to the volume. This requires different infrastructures in which computational power and storage is combined. When big data was a new concept, such technologies basically did not exist. Nowadays, there are many ways to implement such infrastructures, e.g., with Hadoop, Spark, Kafka, Cassandra, HBase, and many others.

1.1.5 Insights, decision making, and process automation

The final part of the definition means that having large amounts of data is not yet big data. Data can only become big data, if it is actually used, e.g., to generate insights, guide decision making, or even automated parts of a business process. This aspect is so important, that there are also definitions of big data in which there is an additional V for *value*.

1.1.6 More Vs

We use a definition with three Vs for big data. Using words that start with V is so popular for big data, that there were multiple suggestions to extend the definition with additional Vs, with up to 42 Vs. Obviously, this is too much and was created with the goal to show that more Vs do not mean that we have a better definition for big data. Regardless, for up to Ten Vs, there are more serious definitions. We already met one the additional Vs, the value which is just called differently in our definition. *Veracity* is another important V that deals with the quality of the data. The more data you have, the harder it is to ensure that the data is reliable and the results can actually be reproduced. This is especially important if the data source changes often, e.g., if news outlets or social network data are analyzed. Volume, velocity, variety, veracity and value are the five V definition of Big Data, which is also popular. We do not cover any of the other Vs here.

1.2 Data Science and Business Intelligence

Data science is a relatively new term for which no agreed upon definition has emerged yet. The reason for this is likely two-fold. On the one hand, the term is very generic, i.e., every use of anything related to data that is remotely scientific can be coined as data science. On the other hand, there is a major hype surrounding the term, which means that companies, consulting firms, funding agencies, and public institutions want to advertise with their use and support of data science.

Due to this, we also do not try do find a good and concise definition for data science. Instead, we look at examples for things that fall under the term data scientist and try do understand the differences to a term that was also popular in the industry a couple of years ago, i.e., business intelligence.

1.2.1 What is Data Science?

Data science brings together mathematics, statistics, and computer science with the goal to generate insights and applications from data.

Mathematics plays a foundational role in how we work with data, because a common goal of many data science projects is to find a mathematical description for a certain aspect related of the data. Thus, data science is ultimately about finding mathematical models. However, the impact on mathematics goes beyond just being a "description language" for models about data. Various fields of mathematics are integral parts of the methods we use to determine models, for example, the following.

- *Optimization* deals with the question how optimal solutions for a target function can be found in a space of possible solutions described by constraints. This is often used to optimize the models we derive from data.
- *Stochastics* is used to describe the behavior of random events through random variables and stochastic process. These are the foundation for the theory of machine learning as well as for many applications.
- *Computational geometry* is required for analyzing data that is spatially distributed, e.g., geographically, astronomically, or on the 3D space in front of a car.
- *Scientific computing* is also related to data science, because more and more applications emerge where machine learning and classical scientific computing are used together.

Statistics deals with the analysis of samples of data through the inference of probability distributions that describe the data, time series analysis, and the definition of statistical tests that evaluate if assumptions on the data likely hold. Concrete aspects from statistics that are relevant for data science are, for example, the following.

- *Linear models* are a versatile means to fit linear descriptions to data for the analysis and may also be used for forecasting future values.
- *Inference* is a similar method for describing data, but mostly through probability distributions instead of linear models.

- *Statistical tests* are an important part in the toolbox of any scientists and can be used to determine how well models work, especially if it is likely that observed effects are only random.
- *Time series analysis* exploits structural patterns in temporal data to analyze the internal structure of data over time and may also be used to forecast future values.

The mathematics and statistics would not be actionable on data without computer science. Additionally, theoretical computer science is also part of the foundations of data science. Examples for concepts from computer science that are relevant for data science are the following.

- *Data structures and algorithms* are the foundation of any efficiently implemented algorithm and the understanding of data structures like trees, hash maps, and lists as well as the run time complexity of algorithms enables the understanding and implementation of efficient data science approaches.
- *Information theory* covers the concepts of entropy and mutual information which are important for many algorithms that are used for data analysis.
- *Databases* are the foundation of efficient storage, access, and nowadays even computation with data and SQL is an invaluable skill for any data scientists, that can often even be used with NoSQL databases.
- *Parallel and distributed computing* is a pre-requisite for any Big Data analysis, the scaling of problems to large groups of users, and the efficient implementation of run time extensive algorithms.
- *Artificial intelligence* deals with logical systems and reasoning that can also be applied in modern data science applications. Please note that we explicitly distinguish between artificial intelligence and machine learning in this script. We use the term artificial intelligence for applications like Deep Blue, the rule-based chess system that was the first computer to beat Gary Kasparow in Chess.
- *Software Engineering* is important for any data science approach that should be implemented in a production system, but also for the general management of data science projects.

Finally, there is *machine learning*, which is parts mathematics, parts statistics, and parts computer science, depending on which approaches for learning you want to use. Machine learning tries to infer knowledge from data and generalize this knowledge to other contexts, e.g., through neural networks, support vector machines, decision trees, or similar algorithms.

1.2.2 Examples for Applications

The field of data science is diverse and has many applications in research, industry, and society. Here are six short examples.

- *Alpha Go* is an example for an intelligent self-learning system. A couple of years ago, Alpha Go surprised the world because it came from seemingly nowhere and beat one of the best players of the game of Go. This was surprising, because prior to Alpha Go, computers were on the level of amateurs when it came to go and far away from even being a challenge for professional players. Alpha Go combined classical rule-based artificial intelligence with a self-learning recurrent neural network, to achieve this.
- *Robotics* relies on machine learning to improve how robots move. Boston dynamics is famous for teaching robots a sense of balance by pushing the robots. The robots *learn* how to avoid falling down over time, the same way toddlers learn this.
- *Marketing* and more specifically targeted advertisements in the Internet are a billion-dollar market based on learning which ads are most relevant for users based on their browsing behavior.
- *Medicine* relies more and more on data driven decision support. IBM Watson, who was initially famous because this was the first artificial intelligence that could beat humans in jeopardy, is now being used to help make decisions about cancer treatments. (Although this is not working as well as hoped for.)
- *Autonomous driving* relies on machine learning for different tasks, most importantly the recognition of objects like other cars, bikes, and pedestrians.

1.2.3 Differences to Business Intelligence

In the industry, business intelligence is a related ancestor of data science that has been in use for years. Gartner defines Business Intelligence as "best practices that enable access to and analysis of information to improve and optimize decisions and performance." Consequently, for many organizations data science is just a rebranding of business intelligence. However, a closer look at typical data science applications and business intelligence applications reveals the differences between the terms. The following table compares the typical techniques, data types, and common questions of business intelligence and data science.

	Business Intelligence	Data Science
Techniques	Dashboards, queries, alerts	Optimization, predictive modelling, forecasting
Data Types	Structured, data warehouses	Any kind, often unstructured
Common Questions	What happened? How much did? When did?	What if? What will? How can we?

As can be seen, business intelligence is focused on the analysis and reporting of the past. Data is typically stored in databases, structured and ready to be analyzed. Data science is more or less a superset. Everything from business intelligence may also be coined as data science, however, data science goes beyond that by considering the future. Thus, data science tries to generalize from the data such that forecasts and predictions are possible, which means more complex questions can be answered, e.g., how different scenarios will play out. This allows deeper insights than business intelligence.

1.3 The Skills of Data Scientists

Data scientists are not computer scientists, mathematicians, statisticians, or domain experts. Instead, the perfect data scientist is a combination of all of that.

- Good mathematics skills, especially about optimization and stochastics.
- Statistician with knowledge about regression, statistical tests, and similar techniques.
- Computer science skills, including programming, databases, algorithms, data structures, parallel computing, and ideally also Big data infrastructures.
- Strong knowledge in the intersection of the fields, especially machine learning.
- Enough domain knowledge to understand the data, the questions that must be answered, and how the questions can be answered with the available data.

Soft skills are also important for data scientists. Team work is often required, as data scientists often work at the intersection between domain experts on the one hand, and technical staff on the other hand. The domain experts teach the data scientists about data, the questions that should be answered, and how the outcome of projects should affect future research and/or business processes. The technical staff often takes over at some point when (and if!) projects are operationalized.

Moreover, the data *scientist* should be skeptical and follow the scientific method. This is especially important when dealing with data, to rule out that effects are purely random.

Because this is a very diverse and complex skill set, the proportion of people who can do all of the above is relatively small. Microsoft Research performed a survey with Microsoft employees to determine which tasks related to data science work on. They found that there are nine different types of data scientists.

- *Polymaths* are general purpose data scientists who fit the complete profile described above, i.e., those who can really do it all, from the underlying mathematics to the deployment of big data infrastructures.

- *Data Evangelists* perform data analysis and actively push for the adoption of data driven methods as well as acting on the gained insights.
- *Data Preparers* query existing data platforms and prepare the data for the analysis.
- *Data Shapers* also work on the preparation of the data but also analyze the data.
- *Data Analyzers* use already prepared data and analyze the data to generate insights.
- *Platform Builders* collect data as well as create and administrate platforms both for the collection and analysis of the data.
- *Moonlighters 50% / Moonlighters 20%* are part-time data scientists, that contribute to data science projects but only in a fraction of their overall work.
- *Insight Actors* use the outcome from data science projects and act on the insights.

CHAPTER

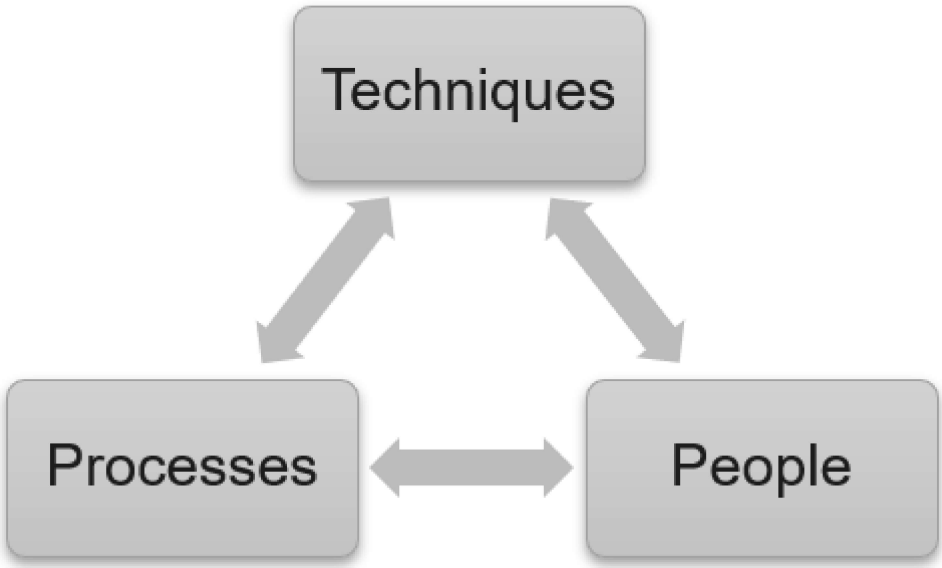
TWO

THE PROCESS OF DATA SCIENCE PROJECTS

2.1 Generic Process Model

2.1.1 Processes

Processes are at the core of any activity, even though we are often not even aware of that. Activities are executed by *people* who apply *techniques*. The *process* guides and organizes the activities of the people and describes the techniques that are used.



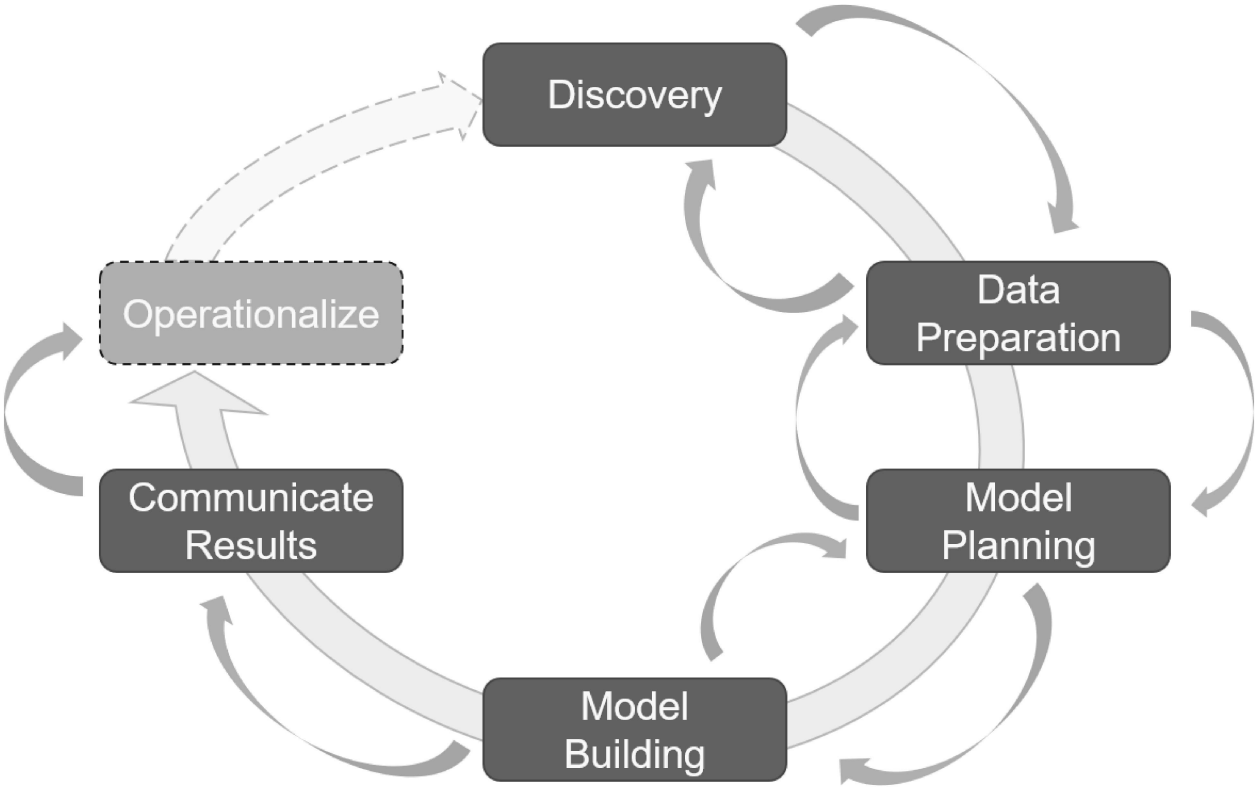
The goal of a good process model is to support the people, e.g., by ensuring that important activities are not forgotten and the recommendation of suitable tools for the solution of problems. In general, process models describe *best practices* that should be applied due to their past success. Through this, the reliance on the existing knowledge and skills of the people should be reduced with the aim to also reduce the risk of project failures. Processes must be supported by the people that use them. If the people do not accept a process, this can hinder productivity and increase the risk of project failures. To ensure that this is not the case, processes should have a measurable positive effect.

A good process requires that the people receive the necessary training for the techniques that should be applied. Moreover, the techniques must be suited for the project. In general, there is no "one fits all" process, because different

aspects influence the choice of processes and techniques, e.g., the project size, whether there are safety-critical aspects, and prior knowledge of the people.

2.1.2 Overview of the Generic Process of Data Science Projects

Our generic process does not prescribe specific techniques, but only the general phases of a project, i.e., a rough sketch of the required activities. For data science projects, there are six generic phases as is shown below.



The process is iterative, i.e., there may be multiple repetitions of the phases in a project. Within one iteration, it is only possible to jump back to prior phases, until the results are communicated. The reason for this is obvious. At this point you decide that these are your results and you communicate them to a broader audience, e.g., the upper management, your customers, other researchers in form of a publication or the submission of your thesis. In the following, we consider each phase in detail.

Discovery

The discovery is the initial phase of the project. The main goal of the discovery is to understand the domain, objectives, data, and to decide if the project has sufficient resources to go forward. To achieve this, many activities must be performed.

The data scientists must *acquire the required knowledge about the domain*, in which the project is conducted. This means that the data scientists must understand the use case associated with the project. Domain experts often collaborate with the data scientists and provide the necessary explanations, e.g., in form of documents or through interviews and workshops. As part of this, the data scientists also gain the required knowledge about the data, i.e., a first and vital understanding about the available information assets that will be used for the analysis in the project. The gained knowledge helps the data scientists to understand the project, as well as with the interpretation of the project results.

Part of the learning about the domain should also be a *consideration of the past*. For academic projects, this is standard practice, as the related work must always be reviewed and considered carefully for any project. However, this is also valuable for work in the industry. Possibly, similar projects were attempted in the past. If this is the case, the results - both positive and negative - from the past projects are invaluable, as they help to avoid similar mistakes and provide guidance about working solutions. Within the bounds of the copyright and patent law, an analysis of the solutions of competitors may also help to better understand the problem as well as potential solutions.

Once the data scientist gained sufficient knowledge about the project, she can start to *frame the problem*. This means that the problem that shall be solved is framed as a data analysis problem. This is different from the goal of the project, which is usually a general business or research objective. The previously acquired domain knowledge is invaluable for this, as the data scientist must understand why the problem is important for the customer in order to frame it correctly. Typical questions that the data scientist must answer for this are, for example, who the stakeholders are and what their interest in the project is. The data scientists learn the current problems (pain points) as well as the goals of the stakeholders from this analysis. Based on this assessment, the objectives of the project can be clearly formulated, as well as how the success of the project will be determined. However, data scientists should not only think about the success, but also about risks that may lead to project failure by missing the objectives.

As part of all of the above, the data scientists learn about the data that may be used for the project. The data may already be readily available, e.g., in a data warehouse. However, it is also possible that data must be collected. In either case, the data scientist must get initial knowledge about the scope and structure of the data and gain a high-level understanding of the available information. Otherwise, a subsequent assessment of the required resources would not be possible.

The *science* part of data science should also not be neglected during the discovery. This means that data analysis should not be purely exploratory, but that clear expectations in form of hypotheses should be formulated that can be tested. Otherwise, there is a high chance that results of the projects do not generalize. Moreover, these hypotheses guide the subsequent phases of the project, especially the model planning and model building. These hypotheses should be discussed with domain experts.

Once the project is completely understood, the final step of the discovery is to decide whether to go forward with the project or not. This assessment should be done based on the risk assessment, as well as on whether the available resources are sufficient for execution of the project. At least the following resources should be considered:

- Technological resources, including resources for the data storage, computational resources, and possibly also whether the required software licenses are available or can be bought.
- The required data, i.e., if the required data is available or can be reasonable collected within the scope of the project. This assessment should look at two dimensions, i.e., the number of data points of the data that must be sufficient to achieve the objectives, as well as the information available for each data point is sufficient. Please note that the collection of additional data should be considered during the assessment of the project risks.
- The available time, both in terms of calendar time and person months. Calendar time is the duration of the project. For projects with a calendar time less than one year, the months in which the project is executed should be considered, as holiday seasons may significantly reduce the availability of personal. This further depends on the geographic distribution of the project team, as different times are critical in different countries (e.g., Lunar new year, Christmas holidays, or more or less the complete August in some countries). Person months are an estimate for the effort that developers and data scientists spent on the project. However, we note that two persons working for one year are usually not twice as productive in a project as a single person, which should be taken into account. This phenomenon is well-known and described in The Mythical Man-Month.
- Human resources, i.e., the concrete personal that should work on the project, including whether the skill set of the personal matches the requirements of the projects.

Projects should only be started if all required resources are available.

Example

Your customer is the owner of a Web shop that sells clothing. They want to increase their sales through cross-selling and ask you to design a solution for this based on data about their past sales. As part of the

discovery, you may do the following:

- You interview the customers to better understand if they already have an idea how they want to increase the cross-sell. You find out that they want to place advertisements for additional products whenever something is added to the basket. This information is vital for you, as other solutions could also have been based on Email advertisements.
- You check other Web shops and look at their solutions.
- You frame the problem to predict which advertisements should be placed based on past shopping behavior of the current customer, past shopping behavior of all customers, and the current content of the shopping basket.
- You identify two relevant stakeholders. 1) The owner of the Web shop, who wants to increase the sales. 2) The customers of the Web shop who want to buy relevant products and have a good user experience. Irrelevant advertisements may lead to a decrease in user experience, while relevant advertisements may even improve the user experience.
- You do not identify relevant pain points in the current operation. The goal is not to solve an existing problem but only the optimization of the revenue of the Web shop.
- From the above, you identify two objectives:
- Increase the number of sales.
- Improve user experience through the placement of relevant products only.
- You will check the objectives by an evaluation of the increase in revenue through predictions and an evaluation of the customer satisfaction. The project is successful if the revenue increases by at least 5% and the customer satisfaction does not decrease. A drop of the customer satisfaction which reduces the revenue is the main risk of the project.
- The available data are mainly customer transactions, i.e., which products where bought together by customers including the data of shopping. The data is stored in a relational database. Other data is not available.
- You formulate three hypotheses. 1) Products which were frequently bought together in the past, will be frequently bought together in the future. 2) There are seasonal patterns in the sales (e.g., summer clothing, winter clothing), which are relevant for the recommendations. 3) The category to which items belong is relevant for the cross-sale, especially the brand and the type of clothing.
- You find that the resources that are available are sufficient for a pilot study that evaluates the feasibility of such predictions for cross-sell. However, an assessment of how this will affect the user experience as well as a roll-out into production cannot be achieved with the resources available and would have to be done in a separate project.

Data Preparation

After the discovery the technical work on the project starts with the data preparation. The data preparation has two major goals: 1) the preparation of the infrastructure for the data analysis and the loading of all relevant data into that infrastructure; and 2) gaining an in-depth understanding of the data.

The effort for the preparation of the infrastructure is somewhere between writing a few lines of code and a huge effort that consumes several person years of resources. If the data is relatively small and easy to access, e.g., through a single SQL query or by loading the data from a comma separated value file, this is trivial. However, if you are dealing with big data, if you also have to collect the data, or if the access to the data is difficult due to some other reason (e.g., data privacy concerns), this can be quite difficult and may require lots of effort.

The general process for getting the data into the infrastructure is called *ETL*: extract, transform, load. First, the data is extracted from where ever it is currently stored. This means writing code for loading data from files, databases,

or potentially collecting the data from other sources through tools, e.g., through Web scraping. Once the data is extracted it is transformed into the required format. This transformation usually includes quality checks, e.g., to filter data with missing values or data that contains implausible values that are likely wrong. Moreover, data must often be (re-)structured and converted into different formats. For example, content from blog posts may have to be split into different fields, e.g., title, content, and comments, character encodings may have to be harmonized, and time stamps might need to be converted into a common format. Once all this is done, the data can be loaded into the analysis environment.

A variant of ETL is to switch the transformation and the loading, i.e., *ELT*. In this case, the raw data that is extracted and loaded directly into the analysis environment and all subsequent transformation are already performed inside the analysis environment. Whether ETL or ELT is a better choice depends on the use case. A common argument for using ELT instead of ETL is that the transformations may be so complex, that they require the computational power of the analysis environment. Moreover, ELT allows the evaluation how different transformations influence the results, because they can be changed flexibly, including access of the raw data for the analysis. A common argument for ETL over ELT is that transformations may be too time consuming to perform them possibly repeatedly after reloading the data.

The second major aspect of the data preparation is to get an in-depth understanding of the data. For this, the data scientists must study the available documentation about the data and apply the domain knowledge to understand what the data means. Ideally, the data scientists know the meaning for every single type of data there is, e.g., every column in a relational database, or what different document types there are in a text mining problem and how a structure can be imposed on this unstructured data. This type of work can be categorized as understanding the *meta data*, i.e., the data about the data.

However, the data should also be considered directly, i.e., the data should be *explored* - an activity that is tightly coupled with the transformations of ETL. This means that data scientists should consider relevant statistical markers and visualize data (Chapter 3). The goal is, e.g., to understand the distribution of numeric data, identify invalid data, determine and remove differences in scales of the data for further harmonization. Additionally, data scientists should try to identify which data they actually need and which data may be removed. While dropping irrelevant data early carries the risk that data is dropped that may actually be useful, it can also be of great help if the volume of the data is reduced significantly. Data scientists should always assess this trade-off.

At the end of the data preparation, all relevant data should be available in the analysis environment and relevant pre-processing steps to transform the data into the required representation should have been performed.

Example (continued)

The sales data is stored in a relational database and consists of 352,152 transactions. Each transaction has on average 2.3 items that were bought and is associated with a time stamp in the ISO 8601 format, as well as the anonymized identifier of the user who bought the items. A separate table stores additional information about the items, e.g., the price, as well as a list of categories to which the item belongs (e.g., male clothing, female clothing, trousers, sweater, socks, brand). There is also additional data available, e.g., the payment type, which you decide to drop for your analysis because you do not expect a reasonable relationship to cross-sell.

The overall volume of the data is about one Gigabyte. You decide to use an ELT process, because loading the from the database only requires about one minute and you can then flexibly explore the data while you define the required transformations.

During the data exploration you identify 2,132 transactions without items, which you drop because these are invalid data. Moreover, you note that certain brands are bought very infrequently. You decide to merge all these brands into a new category “Other brand”.

You decide to create four different representations of the transactions to facilitate using different information in the downstream analysis:

- The items as is.
- The items replaced with the type of the clothing (socks, . . .).