Phase 2: Data Preparation

The second phase of the Data Analytics Lifecycle involves data preparation, which includes the steps to explore, preprocess, and condition data prior to modeling and analysis. In this phase, the team needs to create a robust environment in which it can explore the data that is separate from a production environment. Usually, this is done by preparing an analytics sandbox. To get the data into the sandbox, the team needs to perform ETLT, by a combination of extracting, transforming, and loading data into the sandbox. Once the data is in the sandbox, the team needs to learn about the data and become familiar with it. Understanding the data in detail is critical to the success of the project. The team also must decide how to condition and transform data to get it into a format to facilitate subsequent analysis. The team may perform data visualizations to help team members understand the data, including its trends, outliers, and relationships among data variables. Each of these steps of the data preparation phase is discussed throughout this section. Data preparation tends to be the most labor-intensive step in the analytics lifecycle. In fact, it is common for teams to spend at least 50% of a data science project's time in this critical phase. If the team cannot obtain enough data of sufficient quality, it may be unable to perform the subsequent steps in the lifecycle process. Figure 2-4 shows an overview of the Data Analytics Lifecycle for Phase 2. The data preparation phase is generally the most iterative and the one that teams tend to underestimate most often. This is because most teams and leaders are anxious to begin analyzing the data, testing hypotheses, and getting answers to some of the questions posed in Phase 1. Many tend to jump into Phase 3 or Phase 4 to begin rapidly developing models and algorithms without spending the time to prepare the data for modeling. Consequently, teams come to realize the data they are working with does not allow them to execute the models they want, and they end up back in Phase 2 anyway.

2.3.1 Preparing the Analytic Sandbox

The first subphase of data preparation requires the team to obtain an analytic sandbox (also commonly referred to as a ***workspace***), in which the team can explore the data without interfering with live production databases. Consider an example in which the team needs to work with a company's financial data. The team should access a copy of the financial data from the analytic sandbox rather than interacting with the production version of the organization's main database, because that will be tightly controlled and needed for financial reporting. When developing the analytic sandbox, it is a best practice to collect all kinds of data there, as team members need access to high volumes and varieties of data for a Big Data analytics project. This can include everything from summary-level aggregated data, structured data, raw data feeds, and unstructured text data from call logs or web logs, depending on the kind of analysis the team plans to undertake. This expansive approach for attracting data of all kind differs considerably from the approach advocated by many information technology (IT) organizations. Many IT groups provide access to only a particular subsegment of the data for a specific purpose. Often, the mindset of the IT group is to provide the minimum amount of data required to allow the team to achieve its objectives. Conversely, the data science team wants access to everything. From its perspective, more data is better, as oftentimes data science projects are a mixture of purpose-driven analyses and experimental approaches to test a variety of ideas. In this context, it can be challenging for a data science team if it has to request access to each and every dataset and attribute one at a time. Because of these differing views on data access and use, it is critical for the data science team to collaborate with IT, make clear what it is trying to accomplish, and align goals. During these discussions, the data science team needs to give IT a justification to develop an analytics sandbox, which is separate from the traditional IT-governed data warehouses within an organization. Successfully and amicably balancing the needs of both the data science team and IT requires a positive working relationship between multiple groups and data owners. The payoff is great. The analytic sandbox enables organizations to undertake more ambitious data science projects and move beyond doing traditional data analysis and Business Intelligence to perform more robust and advanced predictive analytics. Expect the sandbox to be large. It may contain raw data, aggregated data, and other data types that are less commonly used in organizations. Sandbox size can vary greatly depending on the project. A good rule is to plan for the sandbox to be at least 5–10 times the size of the original datasets, partly because copies of the data may be created that serve as specific tables or data stores for specific kinds of analysis in the project. Although the concept of an analytics sandbox is relatively new, companies are making progress in this area and are finding ways to offer sandboxes and workspaces where teams can access datasets and work in a way that is acceptable to both the data science teams and the IT groups.

2.3.2 Performing ETLT

As the team looks to begin data transformations, make sure the analytics sandbox has ample bandwidth and reliable network connections to the underlying data sources to enable uninterrupted read and write. In ETL, users perform extract, transform, load processes to extract data from a datastore, perform data transformations, and load the data back into the datastore. However, the analytic sandbox approach differs slightly; it advocates extract, load, and then transform. In this case, the data is extracted in its raw form and loaded into the datastore, where analysts can choose to transform the data into a new state

or leave it in its original, raw condition. The reason for this approach is that there is significant value in preserving the raw data and including it in the sandbox before any transformations take place. For instance, consider an analysis for fraud detection on credit card usage. Many times, outliers in this data population can represent higher-risk transactions that may be indicative of fraudulent credit card activity. Using ETL, these outliers may be inadvertently filtered out or transformed and cleaned before being loaded into the datastore. In this case, the very data that would be needed to evaluate instances of fraudulent activity would be inadvertently cleansed, preventing the kind of analysis that a team would want to do. Following the ELT approach gives the team access to clean data to analyze after the data has been loaded into the database and gives access to the data in its original form for finding hidden nuances in the data. This approach is part of the reason that the analytic sandbox can quickly grow large. The team may want clean data and aggregated data and may need to keep a copy of the original data to compare against or look for hidden patterns that may have existed in the data before the cleaning stage. This process can be summarized as ETLT to reflect the fact that a team may choose to perform ETL in one case and ELT in another. Depending on the size and number of the data sources, the team may need to consider how to parallelize the movement of the datasets into the sandbox. For this purpose, moving large amounts of data is sometimes referred to as Big ETL. The data movement can be parallelized by technologies such as Hadoop or MapReduce, which will be explained in greater detail in Chapter 10, "Advanced Analytics—Technology and Tools: MapReduce and Hadoop." At this point, keep in mind that these technologies can be used to perform parallel data ingest and introduce a huge number of files or datasets in parallel in a very short period of time. Hadoop can be useful for data loading as well as for data analysis in subsequent phases. Prior to moving the data into the analytic sandbox, determine the transformations that need to be performed on the data. Part of this phase involves assessing data quality and structuring the datasets properly so they can be used for robust analysis in subsequent phases. In addition, it is important to consider which data the team will have access to and which new data attributes will need to be derived in the data to enable analysis. As part of the ETLT step, it is advisable to make an inventory of the data and compare the data currently available with datasets the team needs. Performing this sort of gap analysis provides a framework for understanding which datasets the team can take advantage of today and where the team needs to initiate projects for data collection or access to new datasets currently unavailable. A component of this subphase involves extracting data from the available sources and determining data connections for raw data, online transaction processing (OLTP) databases, online analytical processing (OLAP) cubes, or other data feeds. Application programming interface (API) is an increasingly popular way to access a data source [8]. Many websites and social network applications now provide APIs that offer access to data to support a project or supplement the datasets with which a team is working. For example, connecting to the Twitter API can enable a team to download millions of tweets to perform a project for sentiment analysis on a product, a company, or an idea. Much of the Twitter data is publicly available and can augment other datasets used on the project.

2.3.3 Learning About the Data

A critical aspect of a data science project is to become familiar with the data itself. Spending time to learn the nuances of the datasets provides context to understand what constitutes a reasonable value and expected output versus what is a surprising finding. In addition, it is important to catalog the data sources that the team has access to and identify additional data sources that the team can leverage but perhaps does not have access to today. Some of the activities in this step may overlap with the initial investigation of the datasets that occur in the discovery phase. Doing this activity accomplishes several goals. • Clarifies the data that the data science team has access to at the start of the project • Highlights gaps by identifying datasets within an organization that the team may find useful but may not be accessible to the team today. As a consequence, this activity can trigger a project to begin building relationships with the data owners and finding ways to share data in appropriate ways. In addition, this activity may provide an impetus to begin collecting new data that benefits the organization or a specific long-term project. • Identifies datasets outside the organization that may be useful to obtain, through open APIs, data sharing, or purchasing data to supplement already existing datasets

2.3.4 Data Conditioning

Data conditioning refers to the process of cleaning data, normalizing datasets, and performing transformations on the data. A critical step within the Data Analytics Lifecycle, data conditioning can involve many complex steps to join or merge datasets or otherwise get datasets into a state that enables analysis in further phases. Data conditioning is often viewed as a preprocessing step for the data analysis because it involves many operations on the dataset before developing models to process or analyze the data. This implies that the data-conditioning step is performed only by IT, the data owners, a DBA, or a data engineer. However, it is also important to involve the data scientist in this step because many decisions are made in the data conditioning phase that affect subsequent analysis. Part of this phase involves deciding which aspects of particular datasets will be useful to analyze in later steps. Because teams begin forming ideas in this phase about which data to keep and which data to transform or discard, it is important to involve multiple team members in these decisions. Leaving such decisions to a single person may cause teams to return to this phase to retrieve data that may have been discarded. As with the previous example of deciding which data to keep as it relates to fraud detection on credit card usage, it is critical to be

thoughtful about which data the team chooses to keep and which data will be discarded. This can have far-reaching consequences that will cause the team to retrace previous steps if the team discards too much of the data at too early a point in this process. Typically, data science teams would rather keep more data than too little data for the analysis. Additional questions and considerations for the data conditioning step include these. ● What are the data sources? What are the target fields (for example, columns of the tables)? ● How clean is the data? How consistent are the contents and files? Determine to what degree the data contains missing or inconsistent values and if the data contains values deviating from normal. ● Assess the consistency of the data types. For instance, if the team expects certain data to be numeric, confirm it is numeric or if it is a mixture of alphanumeric strings and text. ● Review the content of data columns or other inputs, and check to ensure they make sense. For instance, if the project involves analyzing income levels, preview the data to confirm that the income values are positive or if it is acceptable to have zeros or negative values. ● Look for any evidence of systematic error. Examples include data feeds from sensors or other data sources breaking without anyone noticing, which causes invalid, incorrect, or missing data values. In addition, review the data to gauge if the definition of the data is the same over all measurements. In some cases, a data column is repurposed, or the column stops being populated, without this change being annotated or without others being notified.

2.3.5 Survey and Visualize

After the team has collected and obtained at least some of the datasets needed for the subsequent analysis, a useful step is to leverage data visualization tools to gain an overview of the data. Seeing high-level patterns in the data enables one to understand characteristics about the data very quickly. One example is using data visualization to examine data quality, such as whether the data contains many unexpected values or other indicators of dirty data. (Dirty data will be discussed further in Chapter 3.) Another example is skewness, such as if the majority of the data is heavily shifted toward one value or end of a continuum. Shneiderman [9] is well known for his mantra for visual data analysis of "overview first, zoom and filter, then details-on-demand." This is a pragmatic approach to visual data analysis. It enables the user to find areas of interest, zoom and filter to find more detailed information about a particular area of the data, and then find the detailed data behind a particular area. This approach provides a high-level view of the data and a great deal of information about a given dataset in a relatively short period of time. When pursuing this approach with a data visualization tool or statistical package, the following guidelines and considerations are recommended. ● Review data to ensure that calculations remained consistent within columns or across tables for a given data field. For instance, did customer lifetime value change at some point in the middle of data collection? Or if working with financials, did the interest calculation change from simple to compound at the end of the year? ● Does the data distribution stay consistent over all the data? If not, what kinds of actions should be taken to address this problem? ● Assess the granularity of the data, the range of values, and the level of aggregation of the data. ● Does the data represent the population of interest? For marketing data, if the project is focused on targeting customers of child-rearing age, does the data represent that, or is it full of senior citizens and teenagers? ● For time-related variables, are the measurements daily, weekly, monthly? Is that good enough? Is time measured in seconds everywhere? Or is it in milliseconds in some places? Determine the level of granularity of the data needed for the analysis, and assess whether the current level of timestamps on the data meets that need. ● Is the data standardized/normalized? Are the scales consistent? If not, how consistent or irregular is the data? ● For geospatial datasets, are state or country abbreviations consistent across the data? Are personal names normalized? English units? Metric units? These are typical considerations that should be part of the thought process as the team evaluates the datasets that are obtained for the project. Becoming deeply knowledgeable about the data will be critical when it comes time to construct and run models later in the process.

2.3.6 Common Tools for the Data Preparation Phase

Several tools are commonly used for this phase: ● Hadoop [10] can perform massively parallel ingest and custom analysis for web traffic parsing, GPS location analytics, genomic analysis, and combining of massive unstructured data feeds from multiple sources. ● Alpine Miner [11] provides a graphical user interface (GUI) for creating analytic workflows, including data manipulations and a series of analytic events such as staged data-mining techniques (for example, first select the top 100 customers, and then run descriptive statistics and clustering) on Postgres SQL and other Big Data sources. ● OpenRefine (formerly called Google Refine) [12] is "a free, open source, powerful tool for working with messy data." It is a popular GUI-based tool for performing data transformations, and it's one of the most robust free tools currently available. ● Similar to OpenRefine, Data Wrangler [13] is an interactive tool for data cleaning and transformation. Wrangler was developed at Stanford University and can be used to perform many transformations on a given dataset. In addition, data transformation outputs can be put into Java or Python. The advantage of this feature is that a subset of the data can be manipulated in Wrangler via its GUI, and then the same operations can be written out as Java or Python code to be executed against the full, larger dataset offline in a local analytic sandbox. For Phase 2, the team needs assistance from IT, DBAs, or whoever controls the Enterprise Data Warehouse (EDW) for data sources the data science team would like to use.