CHAPTER

18

A Data Preparation Cookbook

PREAMBLE

In previous chapters, we focused on the performance of various major aspects of predictive analytic modeling and showed how this process can be used in several analytic applications. Novice modelers might consider taking shortcuts in performing the tasks listed in this chapter, but that is likely to lead to the development of an inadequate model or to no good model at all. It is a commonly held maxim in predictive analytics that 60%–90% of the project time will be consumed by data preparation tasks. In this chapter, we will put the "pieces" together for the bulk of the analytic project to show a step-by-step operation in the data preparation phase of CRISP-DM, which may be the most critical part of the project. This chapter will discuss some of the tasks in phases of the CRISP-DM process model, because these tasks are often part of data preparation.

INTRODUCTION

The question arises often with novice analysts about the proper process to follow in a predictive analytics project. The big-picture answer is expressed by the CRISP-DM process model, described in Chapter 3. The primary phases involved with data preparation are business understanding, data understanding, data preparation, and modeling. But why is modeling phase included in a discussion of data preparation? It is because preliminary modeling is a *part* of the normal sequence of events in successful data preparation. This might be considered as a circular and iterative process designed to increase the merits of the model until an "adequate model" is achieved. Thus, the CRISP-DM process model appears as a sequence of processes organized in an iterative processing format, but under the "covers," it has many feedback loops between the sequence of phases in the project. This chapter will present a common sequence of operations in building a customer relationship management (CRM) model. You can adapt this approach to fit any analytic project application. At the end of the chapter, we will present some of the errors committed commonly in the data preparation process to provide further guidance to help you perform successful analytic projects. You might review the discussion of the CRISP-DM process model described in Chapter 3 before proceeding to implement the tasks listed in this cookbook. Also, it is recommended very highly

https://doi.org/10.1016/B978-0-12-416632-5.00018-9

that you compose the tasks in the cookbook for data preparation to form a project plan in some planning tool like Microsoft Project. *The goal is to "plan the work" and then "work the plan."*

CRISP-DM—BUSINESS UNDERSTANDING PHASE

Before you do any data or analytic task, you must understand clearly and define the business goal(s) of the project. It is a *big* mistake to proceed ahead without these goals set clearly in mind. Otherwise, you will be guilty of the target shooters' common mistake of following a sequence of ready-fire-aim. Before we "fire" off the data and analytic processes, we have to know clearly what goals the projects aim to achieve. This is the most important piece of initial "data" to prepare for the successful completion of the project. Tasks are listed in the approximate order that they might be implemented most efficiently. The order of these tasks, however, might change for other projects.

Major tasks in this phase include the following:

Task: Define the Business Goals of the Project

What are the questions you want to answer, and do the answers correspond with the business needs that are driving the project? Proceeding ahead without the answer to this question will lead almost certainly to the failure of the project.

Task: Identify the Business Stakeholders

If you try to produce a product from the analytic project that will affect the operations of various staff functions in the company, you had better include their representatives in the project discussions from the very beginning. Failure to do that may "tick off" the staff members in these functionalities (e.g., marketing) and make it very awkward and time-consuming to implement.

Task: Define Working Relationships

In a CRM analytic project, the two most important functionalities in the company are marketing and information technology (IT). Representatives of IT are the local "owners" of the data you will want to access, and they will have a strong proprietary interest in it. IT staff are necessary in any project to explain what data fields exist, where they are, what restrictions are associated their use, and how to extract them. The analytic modeler must develop good relations with IT staff; otherwise, an adversarial relationship between the modeler and IT staff is likely to develop. They are your "food sources," and you don't want to close down the distribution channel.

Task: Define the Analytical Goals of the Project

The analytic goals of the project must be formed initially in discussions with all stakeholders and defined clearly in their minds and the mind of the modeler. Otherwise, you are likely to discover too late that "you can't get there from here" following the project plan.

Task: Define Acceptance Criteria

You determine at the beginning of the project how you can "know that we are good," both from an analytic standpoint and in the view of the stakeholder also. Failure to do this may delay the project deliverables significantly, as processes and tasks must realigned near the end of the project.

Task: Define the Service Level Agreement (SLA) of the Project

This requirement is very easy to overlook. The marketers and IT staff may be very confident that the model to be developed will be very useful in the company. The implementers of the model, however, might be unable to use it in the form provided. For example, a customer churn project for a large Midwestern bank chain required that the bank relationship managers have 2 weeks to contact high-probability churners to convince them to stay in the bank after it switched from free checking to fee-based checking. That requirement prevented the analyst from using any temporal data as predictors in the model that were less than 2 weeks old (e.g., number of past weekly deposits and withdrawals), lest the customers churned before the managers could contact them. The model delivered required a 2-week forecast horizon to satisfy this SLA, or it would be unacceptable to the bank managers.

Task: Define the Target Variable of the Analysis

The target variable may appear to the analysts and the marketing/IT staff to be appropriate, but it may not satisfy the requirements of the upper management. For example, the target variable for a customer churn model for a South American wireless phone company had to be defined as target = 1 if there was a greater than 70% decline in minutes of use (MOU) over the previous two billing periods, else as target = 0. The upper management wanted a model that would be more diagnostic of the development of the churn "signal" in their records rather than just focusing the fact of churn among past customers.

Task: Create the Project Plan With Milestones and an Appropriate Timeline

Project planning may be attempted by those not formally trained in the art, but this is almost always a mistake. In addition to a data analyst on the modeling team, there should be a project manager, who is adept in project planning, preferable with a Project Management Professional (PMP) certificate from the Project Management Institute (PMI). PMI professionals are trained to include all necessary elements in a project plan and to avoid process mistakes that can prove very costly and time-consuming.

CRISP-DM—DATA UNDERSTANDING PHASE

After the business aspects of the analytic project are worked out and documented clearly, attention can be turned to identifying, accessing, integrating, and understanding of the available data. The major tasks in this phase of the modeling project include the following tasks:

Task: Access Your Data

This task is composed of a number of subtasks that must be orchestrated to produce the customer analytic record (CAR). Not all of these subtasks will be appropriate for all modeling projects:

- Identify existing data sources.
- Determine data ownership issues.
- Determine practical problems with data extraction.
- Evaluate ODBC versus query-based data extraction.
- Identify and resolve any data security issues.
- Identify any data latency issues.
- Define specific data source documents/tables from which data will be extracted.
- Write the data extract scripts to assemble the data from data sources.
- Integrate data sources (files, tables, spreadsheets, etc.).
- Assemble the analytic record with one record per modeling entity (e.g., customer).
- Go-vs-NoGo decision: You must decide whether the nature of the data available is suitable for the analysis of the intended outcome.

Task: Enhancing and Enriching the Data

The data fields provided by company data sources represent only a few of the dimensions of customer information that could be used to define the pattern of the target variable (e.g., customer attrition, or churn). If funds are available to purchase them, there are abundant sources of external information about customers that could prove valuable in defining the target pattern more fully. This value is absolutely necessary for building customer acquisition models, where you have no internal data to use. When internal data are available, incorporating these external data into the CAR is an example of a feedback loop between subtasks in this phase of analysis. These external data sources include a rich variety of information for defining customer locations, attitudes, performance in credit operations, and general public information including the following list:

- Demographic data
- Firmographic data
- Psychographic data
- Credit data
- Census data
- Other public domain data

Task: Characterize and Describe Your Data

Before you begin to use the data elements for any operation, you should describe them and perform several subtask operations on them to diagnose any potential problems associated with their use (e.g., multicollinearity caused by using highly correlated variables in the ensuing analysis).

Common subtasks include the following:

- Calculate descriptive statistics for all variables.
  Information about means, medians, standard deviations, and other metrics can be useful in deciding which variables to use for predictors in the training of the model. For example, the ratio of the mean/standard deviation is a measure of the useful variation in the variable that might be used in the definition of the target variable class or value.
- Categorize input variables into continuous (numbers) and discrete (categories).
  This subtask must be performed to understand what data preparation operations must be performed on each variable. For parametric statistical analysis (which uses only numbers), categorical variables must be converted to “dummy” variables (containing either a 1 or a 0) indicating the presence of a value of a specific category code. This operation will be performed in the data preparation phase.
- Identify/define the target variable(s) in terms of available data elements.
  This subtask may appear to be trivial, but management requirements might define this definition in a rather complicated way.
- Look for outliers, and decide how to handle them in the data preparation phase.
  For some projects, outliers should be recoded to the highest value or the median value, in order to prevent their undue influence of the model solution. In other projects (e.g., fraud analysis), outliers are the target of interest and should be retained.
- For parametric statistical modeling algorithms (e.g., logistic regression), we should perform the following operations:
  Reasons are as follows:
  - Graph data distributions.
  - Determine specific transformation to use for each predictor variable to convert the data distribution to a form as close to the normal curve as possible.
  - Derive any obvious interaction variables.
  - These operations are planned in this phase and performed in the data preparation phase.
- Calculate correlation coefficients.
  This correlation coefficients generated by this subtask can be used to determine which variables are correlated to each other. Only one variable of a group of highly correlated variables should be used; the other one should be deleted.
- Determine which variables have missing data, and design a strategy for missing data handling for each variable with missing values. Missing data imputation will be performed in the data preparation phase.
  This subtask is particularly important, because many machine-learning algorithms used for modeling can delete the entire row of data that contains a missing value in even just one of the variables within that row (case).
- Analysis for data bias
  Types of bias:
  - Sample bias
    Different samples of a larger data population might yield very different analysis results. Make sure that your sample is representative of the population, as far as you can determine.

    - Experimental bias
      A study of the performance of high school students in a certain high school might be biased against dropouts.
    - Measurement bias
      If survey interviewers included deaths that occurred before the time period of the study, it would cause an overestimate of the mortality rate among the study population.
    - Intentional bias
      The problem is more pernicious, because it may be unexpected, and it is very hard to assess. An experimenter in the mortality study above might limit the population interviewed to only those with healthy diets, because of the desire to prove a "point."
- Determine if data set samples (subsamples of the entire dataset) should be performed prior to analysis.
  Reasons:
    - Reduce data volume.
    - Incorporate into any planned resampling or cross validation operations.
    - Develop plans for any undersampling or oversampling strategy to be employed to correct for effects related to rare target values during the data preparation operations.

CRISP-DM—DATA PREPARATION PHASE

This phase of the process model is often the most crucial one to do properly. Because of the diversity of tasks in this phase and the complexity of some of the operations, it may consume up to 90% of the project time.

Task: Perform Any Planned Sampling Regime

Types of sampling regimes are as follows:

- Random sampling
  It is performed to draw a representative group of data rows from a larger data population.
- Stratified sampling
  It is performed when a population is composed of two more groups that can be grouped together according to the same criteria (e.g., geographic location).
- Oversampling and undersampling
  It is performed to produce the same number of rows for each target class. See Chapter 4 for more information on this topic.
- Assign case weights or prior probabilities to specific target classes, instead of balancing data sets with a rare target class.
  Some analytic tools can balance data sets with rare target classes by using differential weights or using probability of occurrence of the target classes (prior probabilities) to govern the effect of the classification operation. See Chapter 4 for more information.

Task: Data Cleaning

Types of cleaning operations are as follows:

- Deletion of "garbage" values
  Data tables from which you extract data might have inappropriate records in them, which cannot be used for modeling (e.g., training records).
- Deletion of sensitive information
  Data sets intended for analysis might contain certain data elements that are confidential or protected by law (e.g., Social Security number). These data elements must be deleted for any analysis it performed.

Task: Data Reduction

Types of data reduction operations are as follows:

- Reduction of dimensionality (of variables or "features," where a feature is defined as a transformed variable). This is one of the most important operations in data preparation. Many analytic packages provide feature reduction tools. There are a number of other methods for reducing the number of features to be submitted to a modeling algorithm. The primary reasons to reduce the number of features in an analysis are (1) to reduce the mathematical complexity of the feature space, enabling the modeling algorithm to work more efficiently, and (2) to reduce the "noise" of the target signal. This subject is discussed in greater detail in Chapter 5.
- Reduction of numerosity
  This operation refers to the reduction of the number of distinct numeric values. A modeling algorithm is trained to recognize a pattern in the numbers or categories of the data set. The efficiency of the process of defining the pattern to the "senses" of the algorithm can be increased significantly in efficiency and speed of operation by reducing the number of distinct numeric values it must process. Neural network algorithms accept categorical inputs, but they must transform the categories to numbers before they can be processed. Decision trees can work directly with category names, but their efficiency in building and trimming branches of the tree can be enhanced by reducing the numerosity also.
  Common approaches are as follows:

- Aggregating
  Data from similar variables can be aggregated to generate a generalization of the group of inputs:
  - Clustering
    Groups of records can be grouped to reduce the data volume.
  - Sampling
    A 10%–20% sample of the data population may provide sufficient rows to build the pattern for the algorithm to recognize.
- Discretization
  The narrow definition of this term covers only the operation of binning of numeric values to move from discrete values along a range of a variable to groups of values in specific

subranges of the variable (e.g., 0–9, 10–19, and 20–29). This term is applied sometimes to the process of derivation of "dummy" variables from categorical variables.

- Development of concept hierarchy generation
  For example, colleges, high schools, middle schools, and grammar schools compose a concept hierarchy for schools. The specific concept in this case might relate to the sophistication of curriculum materials. The fact that a given variable refers to data from high school students might not matter nearly as much as the information about where in the hierarchy of curriculum complexity it stands.

Task: Standardization

Standardization or normalization of data is necessary for parametric statistical algorithms, but not for machine-learning algorithms. The most common form of standardization is to calculate the *Z*-score for each numeric value. This transformation converts the variable to a range between − infinity and + infinity, in which 99.5% of the values (in a normal data distribution) range between −3 and +3 values for the *Z*-score (this range also corresponds to a departure of values within three standard deviations from the mean).

Task: Recoding

Some variables may include codes that refer to two different coding systems. The modeler must recode code values on the old coding schema to the new schema. Another example of the need for recoding is to replace "garbage" data (e.g., data entry errors) with blank filler codes.

Task: Filtering

Use of some rows in a data set might not be legally permissible to use in predictive models. For example, the Gramm-Leach-Bliley Act of 1999 constrains banks from using customer-specific data in public reports, but data summaries across customers are permitted.

Task: Missing Value Imputation

This task is described in detail in Chapter 4; therefore, only short summaries of common techniques used are presented here. There are three types of operations to impute missing values in variables:

- With constants
  This operation fills all missing values for a list of variables in the data stream with a specified constant, either a number or a text code.
- With formulas
  The mathematical evaluation of a formula can be used to fill missing values, if relationships between the missing value and the elements of the formula are known.

- With models
  A simple model can be used to impute missing values, using other variables as predictor. For example, age, home ownership, and zip code might be useful predictors of income.

Task: Derived Variables

- Summarization
  In very detailed data domains (e.g., retail transactions and call detail phone records), the analysis grain of detail is coarser than the raw data grain. Call detail phone records at the telecommunication switch consist of link paths, start and end times of the phone call converted to minutes of use (MOU). If the analysis time grain is specified at the weekly level, the MOUs for all calls in a week must be aggregated for each customer. Another example of data summarization in a telecommunication data domain is to calculate the means and standard deviations for certain variables for each week.
- Complex calculations
  In a Banking data domain, it may be necessary to calculate the distance from prospective customers and the nearest bank branch, using spherical geometry.
- Dummy variables
  Dummy variables are derived for each class of a categorical variable, installing a 1 for a given row is the class is present, else 0. Dummy variables are discussed more fully in Chapter 4.
- Distribution transformations
  Parametric statistical predictive algorithms (e.g., linear regression) assume that the distribution of each variable is normal. The degree to which the data distribution of a variable departs from normality is proportional to the amount of error that may be added to the solution. Various transformation functions (e.g., square, square root, and natural log) can be applied to the data distributions until the transformed data distribution approaches normality.

Task: Handling of Outliers

This task requires the initial decision whether or not to leave outliers in the data set. In many cases, it is preferable to recode outliers to some value (e.g., the largest), to focus the modeling algorithm on the normal range of the variable in its training operation.

Task: Handling of Temporal Data

Temporal data are one of the most challenging problems encountered by analysts. If the goal is to forecast sales, for example, traditional time-series analysis algorithms can be used effectively (e.g., ARIMA). If the goal is to relate a customer response in the future to a series of customer events in the past, these events must be copied to the row in the CAR for a given customer and renamed for the time periods in the past to which they pertain. Such a variable is a temporal abstraction and is called commonly a *lag variable.* Lag variables can be very predictive in many customer data domains, such as telecommunications (where they were used initially in the late 1990s). They can also be effect predictors in insurance, credit, and banking data domains.

CRISP-DM—MODELING PHASE

Task: Preliminary Modeling Operations to Test the Effectiveness of Certain Data Preparation Operations

This task doesn't appear at first consideration to be a part of data preparation. It is, however, one of the most important operations in the data preparation process. There are many data preparation operations on variables (manipulations) and entire data sets (conditionings) that work better than others for a given data set. The only way to find the right operation for a given data set is to try various operations and test their effectiveness with a preliminary model. If a given operation is not effective in a model, it can be dropped from the processing sequence. Preliminary modeling operations do not include model enhancements that might be performed to build the optimum model for a given project.

Examples of data preparation operations that might be tested include the following:

- Standardization
- Undersampling operation to compensate for a rare target class
- Oversampling operation to compensate for a rare target class
- Data segmentation operations

  It was discovered in a wireless phone company that urban customers had a very different calling behavior pattern than rural callers. Separate churn models were built for urban and rural customers. Otherwise, the very different calling behavior signals would confuse the pattern of the target variable in the modeling process.

The cookbook for data preparation presented above may be followed with an approach that is too closely associated with the details of performing the relevant tasks in an analytic project. It is helpful, however, to review some mistakes that are made commonly in analytic projects, with a view in mind to avoid them like the plague while performing the project tasks.

18 COMMON MISTAKES IN DATA PREPARATION IN PREDICTIVE ANALYTICS PROJECTS

The following common mistakes have occurred in one project or another in the past of the authors' experience. The first 10 are ordered roughly in terms of their estimated importance or impact on the fidelity of the results. The last eight are judged to be of lesser importance, yet still they occur often enough to be listed here:

1. *Failure to fill all missing values*

   This can be a very pernicious error when modeling with machine-learning tools, because the algorithm you are using may delete the entire row if there is a missing value in any column. Failure to fill all missing values before modeling might reduce the data in the structure submitted to the algorithm to a level that will not serve to train a good model.
2. *Modeling past response with future data*

   The most common form of this error is including a predictor variable (the "past") that is part of the definition of the target (the "future" response). For example, you can't use an Exit_Reason variable as a predictor if the code pertains to the reason the customer left the company ("churned"). All churn customers will have values in the Exit_Reason variable, and those who are still in the company will not. The goal in a churn model is to predict a future response with data variables from the past, but the Exit_Reason variable describes the future response, and cannot be used to predict its occurrence. Whenever a perfect model is built, it is most likely that some information related to the "future" response has been used to predict that response.

3. *Failure to evaluate variables sufficiently.*

Examples of this type of error include (1) using the wrong method of variable selection, (2) using only one variable selection method, and (3) failure to include the right variables in the short list of variables submitted to the modeling algorithm.

The final list of variables submitted to the modeling algorithm (the short list) is a very important part of the modeling project. This list is very valuable to the project and in the marketplace at large. For example, a small company worked for a year to generate this list from thousands of candidate variables in a large client company. The client refused to pay for it; the small company sued and won a judgment of $17 million for the short list.

4. *Failure to partition data sets into three subsets*

Some analytic tools provide techniques for dividing (partitioning) an input data set into a number of subsets prior to analysis. Some tools generate only two data sets, the training data set and the testing data set. These two partitions are used in the training and model testing processes of many machine-learning algorithms. The performance of the model is defined usually according to how well the model predicts the testing data set. This is not the best way to do it. Alternately, a third data set should be partitioned and used to model evaluation, instead of the testing data set. The reason is that model evaluation with the testing data set is *tautologous.* A tautology is a definition of something in terms of itself. The model building process uses the testing data set to choose the best model, and it should *not* be used to evaluate its performance.

5. *Using the record number or an ID number as a predictor*

This is an easy error to commit. Make sure that all identifying columns of the data set are excluded from the list of predictor variables submitted to the modeling algorithm, including ID, name, and street address. For some applications, however, city name and zip code can be valuable predictor variables (they are surrogates for geographic location).

6. *Failure to consider interactions and deriving new variables to express them*

The assumption in parametric statistical analysis that affects of all predictors is independent. This assumption is violated significantly in many data sets. These interactions might be important predictors. Consider a very simple example of length, width, and height. Independently, they might have very little relationship to a model, but the product of all three (volume) might be very predictive.

7. *Failure to devote enough effort to data transformations and deriving new variables*

Even though certain variables may be used currently and serve as predictor variables in their own right, both logical and mathematical combinations of them can be even more predictive. Consider variables NUMPROM (number of marketing promotions) and NGIFT (number of donations) in 1998 KDD-Cup competition data set (available in the UC-Irvine machine-learning archive). Alone, these variables are valid, but are not important predictors. The ratio of them (NGIFT/NUMPROM) represents the frequency of donation (PROMO_FREQ), and it is very predictive of the binary target for donation.

8. *Forgetting about outliers*

In most applications, values beyond three standard deviations from the mean happen only about 0.5% of the time (in a normal distribution). If a value is far beyond three standard deviations, it is most likely an anomaly or an error. We should not model on errors or anomalies in the data. We can clarify the vast bulk of the target "signal" by removing outliers (by recoding them). Yes, it deletes data, but the benefit of removing the outliers will in most cases far exceed any effects of this data loss. In some cases, outliers are the target of interest (e.g., in fraud and intrusions) and should be retained.

9. *Failure to normalize or standardize numerical inputs for parametric statistical algorithms*

Parametric statistical algorithms require that a number of assumptions be satisfied (e.g., normality, independency, and linearity). Another assumption is that the scales of all predictor variables are relatively similar. If one variable has a scale that is significantly greater than the other variables, the estimation of parameters will be biased toward the parameter with the greater scale. The solution to this problem is to convert the data to a common scale. This can perform with a function. The most common function is to use the *Z*-transform function, which will convert scales to a range of − infinity to + infinity, but 99.5% of the values will lie between −3 and +3. Another approach is to "normalize" the data ranges from 0 to 1.0.

Machine-learning algorithms (e.g., neural nets, decision trees, SVMs, and Bayesian classifiers) don't require standardization, but most models will be improved by using standardized data.

10. *Using biased samples*

A sample of public high school students for the use of drugs can be very biased because private schoolers, homeschoolers, and dropouts are not included in the sample. The resulting sample is biased toward those students that are successful in a public school, and may not be representative of students in the population at large. Policies formed on the basis of the analysis of this biased sample may lead to invalid conclusions in many cases.

Another form of sampling bias is when data are collected for only positive results.

You might receive 100 applications for a service and reject 50 of them initially. But what if the initial screening is faulty? You might expand your analysis to include the 50 who were rejected initially to see if any of them are potentially good customers otherwise. In the credit industry, this operation is called "reject inference." Some of the potential customers rejected initially can be good customers after all.

11. *Ignoring temporal data, and failing to derive temporal abstractions*

It is difficult to relate temporal variables directly to a response variable (the target). Traditional time-series analyses model only the target "signal," and do not use predictor variables. Temporal information can be analyzed only after successive rows of temporal data (in a time-series) are abstracted and related to the target variable. This is performed by deriving "lag" variables, in which the effect of change in the target lags behind the influence of the variable by some time period. See Chapter 14 for more information on this operation.

12. *Using historical data that is not accurate*

For example, customers may have moved from a city location to a suburban location part way through the historical sequence of data for the customer. In this case, the geographic location of all records should be standardized to city or suburban, depending on the goal of the model.

13. *Ignoring historical changes in codes*

This error may appear to be similar to #12 above, but the context is very different. Sometimes, a coding scheme for a given variable may change during the historical course of data collection. This mixture of codes for the same thing must be reconciled before analysis can proceed.

14. *Including variables that have changed in the past, in format or codes, or which are likely to change in the future*

Error #6 above refers to previous coding changes. These changes may happen in the future also. For example, you might be tempted to include in the model a predictor whose data format or code is expected to change in the future. One example might be a predictor in studies of patients in a medical office that expects to change to electronic medical records (EMRs) in the near future.

15. *Failure to remove duplicate records*

The initial error is failing to scan for duplicate records. Some analytic tools provide capabilities for scanning for duplicate records. Another strategy for finding duplicate records is to do some multilevel sorts of your data sets to see if multiple records sort out together and appear to be duplicates.

16. *Including retired fields as predictors*

Occasionally, certain fields in a database might not be used in business anymore. You must be careful to qualify each data field to be included in the list of candidate predictors to make sure that each one is used currently in business processes.

17. *Confusing correlation with causation*

A principle that is drummed into students in their statistics classes is that correlation does not imply causality. A good example of this is seen in a study in Nature magazine in 1999, which showed that children who sleep with a light on are more likely to develop myopia in later life (Quinn et al., 1999). One conclusion that might be made from these data is that sleeping with a light on causes myopia. But a more recent study showed that children of myopic parents had a high likelihood of developing myopia (Ohio State University Research News, 2000). It is possible that myopic parents left lights on in children's rooms to be able to see them while attending them at night. Analysis of business data correlations might follow a similar pattern, and we might be tempted to draw erroneous conclusions about causality.

18. *Thinking that failure is not an option*

Trial and error is one of the most tried and true scientific methods used. One industry management expert promotes the concept of “fast failures” (Peters, 1988).

POSTSCRIPT

The lists of steps in data preparation presented above are listed in the general order in which they should be performed. In practice, however, there is much feedback of information to previous steps in the sequence. Errors also can be made, but not noticed until you perform some later step; you have to go back and fix them. Just like in the CRISP-DM overall process model presented in Chapter 3, data preparation follows an iterative process with feedbacks.

References

Ohio State University Research News, 9 March 2000. Night lights don't lead to nearsightedness, study suggests.

Peters, T., 1988. Thriving on Chaos—Handbook for a Management Revolution. Harper Perennial, New York, NY, 736 pp.

Quinn, G.E., Shin, C.H., Maguire, M.G., Stone, R.A., 1999. Myopia and ambient lighting at night. Nature 399 (6732), 113–114. https://doi.org/10.1038/20094. 10.1038%2F20094. PMID 10335839.