2 Data Mining Process

In order to systematically conduct data mining analysis, a general process is usually followed. There are some standard processes, two of which are described in this chapter. One (CRISP) is an industry standard process consisting of a sequence of steps that are usually involved in a data mining study. The other (SEMMA) is specific to SAS. While each step of either approach isn't needed in every analysis, this process provides a good coverage of the steps needed, starting with data exploration, data collection, data processing, analysis, inferences drawn, and implementation.

CRISP-DM

There is a Cross-Industry Standard Process for Data Mining (CRISP-DM) widely used by industry members. This model consists of six phases intended as a cyclical process (see Fig. 2.1):

- *Business Understanding* Business understanding includes determining business objectives, assessing the current situation, establishing data mining goals, and developing a project plan.
- *Data Understanding* Once business objectives and the project plan are established, data understanding considers data requirements. This step can include initial data collection, data description, data exploration, and the verification of data quality. Data exploration such as viewing summary statistics (which includes the visual display of categorical variables) can occur at the end of this phase. Models such as cluster analysis can also be applied during this phase, with the intent of identifying patterns in the data.
- *Data Preparation* Once the data resources available are identified, they need to be selected, cleaned, built into the form desired, and formatted. Data cleaning and data transformation in preparation of data modeling needs to occur in this phase. Data exploration at a greater depth can be applied during this phase, and additional models utilized, again providing the opportunity to see patterns based on business understanding.

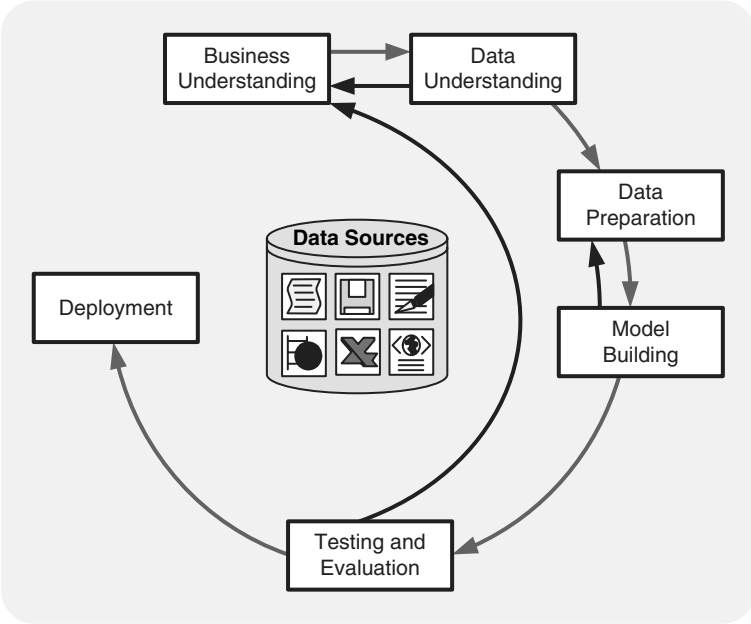

Fig. 2.1. CRISP-DM process

- *Modeling* Data mining software tools such as visualization (plotting data and establishing relationships) and cluster analysis (to identify which variables go well together) are useful for initial analysis. Tools such as generalized rule induction can develop initial association rules. Once greater data understanding is gained (often through pattern recognition triggered by viewing model output), more detailed models appropriate to the data type can be applied. The division of data into training and test sets is also needed for modeling.
- *Evaluation* Model results should be evaluated in the context of the business objectives established in the first phase (business understanding). This will lead to the identification of other needs (often through pattern recognition), frequently reverting to prior phases of CRISP-DM. Gaining business understanding is an iterative procedure in data mining, where the results of various visualization, statistical, and artificial intelligence tools show the user new relationships that provide a deeper understanding of organizational operations.
- *Deployment* Data mining can be used to both verify previously held hypotheses, or for knowledge discovery (identification of unexpected and useful relationships). Through the knowledge discovered in the earlier phases of the CRISP-DM process, sound models can be obtained

that may then be applied to business operations for many purposes, including prediction or identification of key situations. These models need to be monitored for changes in operating conditions, because what might be true today may not be true a year from now. If significant changes do occur, the model should be redone. It's also wise to record the results of data mining projects so documented evidence is available for future studies.

This six-phase process is not a rigid, by-the-numbers procedure. There's usually a great deal of backtracking. Additionally, experienced analysts may not need to apply each phase for every study. But CRISP-DM provides a useful framework for data mining.

Business Understanding

The key element of a data mining study is knowing what the study is for. This begins with a managerial need for new knowledge, and an expression of the business objective regarding the study to be undertaken. Goals in terms of things such as "What types of customers are interested in each of our products?" or "What are typical profiles of our customers, and how much value do each of them provide to us?" are needed. Then a plan for finding such knowledge needs to be developed, in terms of those responsible for collecting data, analyzing data, and reporting. At this stage, a budget to support the study should be established, at least in preliminary terms.

In customer segmentation models, such as Fingerhut's retail catalog business, the identification of a business purpose meant identifying the type of customer that would be expected to yield a profitable return. The same analysis is useful to credit card distributors. For business purposes, grocery stores often try to identify which items tend to be purchased together so it can be used for affinity positioning within the store, or to intelligently guide promotional campaigns. Data mining has many useful business applications, some of which will be presented throughout the course of the book.

Data Understanding

Since data mining is task-oriented, different business tasks require different sets of data. The first stage of the data mining process is to select the related data from many available databases to correctly describe a given business task. There are at least three issues to be considered in the data selection. The first issue is to set up a concise and clear description of the problem. For example, a retail data-mining project may seek to identify

spending behaviors of female shoppers who purchase seasonal clothes. Another example may seek to identify bankruptcy patterns of credit card holders. The second issue would be to identify the relevant data for the problem description. Most demographical, credit card transactional, and financial data could be relevant to both retail and credit card bankruptcy projects. However, gender data may be prohibited for use by law for the latter, but be legal and prove important for the former. The third issue is that selected variables for the relevant data should be independent of each other. Variable independence means that the variables do not contain overlapping information. A careful selection of independent variables can make it easier for data mining algorithms to quickly discover useful knowledge patterns.

Data sources for data selection can vary. Normally, types of data sources for business applications include *demographic data* (such as income, education, number of households, and age), *socio-graphic data* (such as hobby, club membership, and entertainment), *transactional data* (sales records, credit card spending, issued checks), and so on. The data type can be categorized as quantitative and qualitative data. *Quantitative data* is measurable using numerical values. It can be either discrete (such as integers) or continuous (such as real numbers). *Qualitative data*, also known as categorical data, contains both nominal and ordinal data. Nominal data has finite non-ordered values, such as gender data which has two values: male and female. Ordinal data has finite ordered values. For example, customer credit ratings are considered ordinal data since the ratings can be excellent, fair, and bad. Quantitative data can be readily represented by some sort of probability distribution. A probability distribution describes how the data is dispersed and shaped. For instance, normally distributed data is symmetric, and is commonly referred to as bell-shaped. Qualitative data may be first coded to numbers and then be described by frequency distributions. Once relevant data are selected according to the data mining business objective, data preprocessing should be pursued.

Data Preparation

The purpose of data preprocessing is to clean selected data for better quality. Some selected data may have different formats because they are chosen from different data sources. If selected data are from flat files, voice message, and web text, they should be converted to a consistent electronic format. In general, data cleaning means to filter, aggregate, and fill in missing values (*imputation*). By filtering data, the selected data are examined for outliers and redundancies. Outliers differ greatly from the majority

of data, or data that are clearly out of range of the selected data groups. For example, if the income of a customer included in the middle class is $250,000, it is an error and should be taken out from the data mining project that examines the various aspects of the middle class. Outliers may be caused by many reasons, such as human errors or technical errors, or may naturally occur in a data set due to extreme events. Suppose the age of a credit card holder is recorded as "12." This is likely a human error. However, there might actually be an independently wealthy pre-teen with important purchasing habits. Arbitrarily deleting this outlier could dismiss valuable information.

Redundant data are the same information recorded in several different ways. Daily sales of a particular product are redundant to seasonal sales of the same product, because we can derive the sales from either daily data or seasonal data. By aggregating data, data dimensions are reduced to obtain aggregated information. Note that although an aggregated data set has a small volume, the information will remain. If a marketing promotion for furniture sales is considered in the next 3 or 4 years, then the available daily sales data can be aggregated as annual sales data. The size of sales data is dramatically reduced. By smoothing data, missing values of the selected data are found and new or reasonable values then added. These added values could be the average number of the variable (mean) or the mode. A missing value often causes no solution when a data-mining algorithm is applied to discover the knowledge patterns.

Data can be expressed in a number of different forms. For instance, in CLEMENTINE, the following data types can be used.

- *RANGE* Numeric values (integer, real, or date/time).
- *FLAG* Binary – Yes/No, 0/1, or other data with two outcomes (text, integer, real number, or date/time).
- *SET* Data with distinct multiple values (numeric, string, or date/time).
- *TYPELESS* For other types of data.

Usually we think of data as real numbers, such as age in years or annual income in dollars (we would use RANGE in those cases). Sometimes variables occur as either/or types, such as having a driver's license or not, or an insurance claim being fraudulent or not. This case could be dealt with using real numeric values (for instance, 0 or 1). But it's more efficient to treat them as FLAG variables. Often, it's more appropriate to deal with categorical data, such as age in terms of the set {young, middle-aged, elderly}, or income in the set {low, middle, high}. In that case, we could group the data and assign the appropriate category in terms of a string,

using a set. The most complete form is RANGE, but sometimes data does not come in that form so analysts are forced to use SET or FLAG types. Sometimes it may actually be more accurate to deal with SET data types than RANGE data types.

As another example, PolyAnalyst has the following data types available:

- *Numerical* Continuous values
- *Integer* Integer values
- *Yes/no* Binary data
- *Category* A finite set of possible values
- *Date*
- *String*
- *Text*

Each software tool will have a different data scheme, but the primary types of data dealt with are represented in these two lists.

There are many statistical methods and visualization tools that can be used to preprocess the selected data. Common statistics, such as max, min, mean, and mode can be readily used to aggregate or smooth the data, while scatter plots and box plots are usually used to filter outliers. More advanced techniques (including regression analyses, cluster analysis, decision tree, or hierarchical analysis) may be applied in data preprocessing depending on the requirements for the quality of the selected data. Because data preprocessing is detailed and tedious, it demands a great deal of time. In some cases, data preprocessing could take over 50% of the time of the entire data mining process. Shortening data processing time can reduce much of the total computation time in data mining. The simple and standard data format resulting from data preprocessing can provide an environment of information sharing across different computer systems, which creates the flexibility to implement various data mining algorithms or tools.

As an important component of data preparation, data transformation is to use simple mathematical formulations or learning curves to convert different measurements of selected, and clean, data into a unified numerical scale for the purpose of data analysis. Many available statistics measurements, such as mean, median, mode, and variance can readily be used to transform the data. In terms of the representation of data, data transformation may be used to (1) transform from numerical to numerical scales, and (2) recode categorical data to numerical scales. For numerical to numerical scales, we can use a mathematical transformation to “shrink” or “enlarge” the given data. One reason for transformation is to eliminate differences in variable scales. For example, if the attribute “salary” ranges from

"$20,000" to "$100,000," we can use the formula $S = (x - min)/(max - min)$ to "shrink" any known salary value, say $50,000 to 0.6, a number in [0.0, 1.0]. If the mean of salary is given as $45,000, and standard deviation is given as $15,000, the $50,000 can be normalized as 0.33. Transforming data from the metric system (e.g., meter, kilometer) to English system (e.g., foot and mile) is another example. For categorical to numerical scales, we have to assign an appropriate numerical number to a categorical value according to needs. Categorical variables can be ordinal (such as less, moderate, and strong) and nominal (such as red, yellow, blue, and green). For example, a binary variable {yes, no} can be transformed into "1 = yes and 0 = no." Note that transforming a numerical value to an ordinal value means transformation with order, while transforming to a nominal value is a less rigid transformation. We need to be careful not to introduce more precision than is present in the original data. For instance, Likert scales often represent ordinal information with coded numbers (1–7, 1–5, and so on). However, these numbers usually don't imply a common scale of difference. An object rated as 4 may not be meant to be twice as strong on some measure as an object rated as 2. Sometimes, we can apply values to represent a block of numbers or a range of categorical variables. For example, we may use "1" to represent the monetary values from "$0" to "$20,000," and use "2" for "$20,001–$40,000," and so on. We can use "0001" to represent "two-store house" and "0002" for "one-and-half-store house." All kinds of "quick-and-dirty" methods could be used to transform data. There is no unique procedure and the only criterion is to transform the data for convenience of use during the data mining stage.

Modeling

Data modeling is where the data mining software is used to generate results for various situations. A cluster analysis and visual exploration of the data are usually applied first. Depending upon the type of data, various models might then be applied. If the task is to group data, and the groups are given, discriminant analysis might be appropriate. If the purpose is estimation, regression is appropriate if the data is continuous (and logistic regression if not). Neural networks could be applied for both tasks.

Decision trees are yet another tool to classify data. Other modeling tools are available as well. We'll cover these different models in greater detail in subsequent chapters. The point of data mining software is to allow the user to work with the data to gain understanding. This is often fostered by the iterative use of multiple models.

Data Treatment

Data mining is essentially the analysis of statistical data, usually using enormous data sets. The standard process of data mining is to take this large set of data and divide it, using a portion of the data (the *training set*) for development of the model (no matter what modeling technique is used), and reserving a portion of the data (the *test set*) for testing the model that's built. In some applications a third split of data (*validation set*) is used to estimate parameters from the data. The principle is that if you build a model on a particular set of data, it will of course test quite well. By dividing the data and using part of it for model development, and testing it on a separate set of data, a more convincing test of model accuracy is obtained.

This idea of splitting the data into components is often carried to additional levels in the practice of data mining. Further portions of the data can be used to refine the model.

Data Mining Techniques

Data mining can be achieved by Association, Classification, Clustering, Predictions, Sequential Patterns, and Similar Time Sequences.[1]

In *Association*, the relationship of a particular item in a data transaction on other items in the same transaction is used to predict patterns. For example, if a customer purchases a laptop PC (X), then he or she also buys a mouse (Y) in 60% of the cases. This pattern occurs in 5.6% of laptop PC purchases. An association rule in this situation can be "X implies Y, where 60% is the confidence factor and 5.6% is the support factor." When the confidence factor and support factor are represented by linguistic variables "high" and "low," respectively, the association rule can be written in the fuzzy logic form, such as: "where the support factor is low, X implies Y is high." In the case of many qualitative variables, fuzzy association is a necessary and promising technique in data mining.

In *Classification*, the methods are intended for learning different functions that map each item of the selected data into one of a predefined set of classes. Given the set of predefined classes, a number of attributes, and a "learning (or training) set," the classification methods can automatically predict the class of other unclassified data of the learning set. Two key research problems related to classification results are the evaluation of misclassification and prediction power. Mathematical techniques that are often used to construct classification methods are binary decision trees, neural networks, linear programming, and statistics. By using binary

[1] D.L. Olson, Yong Shi (2007). *Introduction to Business Data Mining*, Boston: McGraw-Hill/Irwin.

decision trees, a tree induction model with a "Yes–No" format can be built to split data into different classes according to its attributes. Models fit to data can be measured by either statistical estimation or information entropy. However, the classification obtained from tree induction may not produce an optimal solution where prediction power is limited. By using neural networks, a neural induction model can be built. In this approach, the attributes become input layers in the neural network while the classes associated with data are output layers. Between input layers and output layers, there are a larger number of hidden layers processing the accuracy of the classification. Although the neural induction model often yields better results in many cases of data mining, since the relationships involve complex nonlinear relationships, implementing this method is difficult when there's a large set of attributes. In linear programming approaches, the classification problem is viewed as a special form of linear program. Given a set of classes and a set of attribute variables, one can define a cut-off limit (or boundary) separating the classes. Then each class is represented by a group of constraints with respect to a boundary in the linear program. The objective function in the linear programming model can minimize the overlapping rate across classes and maximize the distance between classes. The linear programming approach results in an optimal classification. However, the computation time required may exceed that of statistical approaches. Various statistical methods, such as linear discriminant regression, quadratic discriminant regression, and logistic discriminant regression are very popular and are commonly used in real business classifications. Even though statistical software has been developed to handle a large amount of data, statistical approaches have a disadvantage in efficiently separating multiclass problems in which a pair-wise comparison (i.e., one class versus the rest of the classes) has to be adopted.

Cluster analysis takes ungrouped data and uses automatic techniques to put this data into groups. Clustering is unsupervised, and does not require a learning set. It shares a common methodological ground with Classification. In other words, most of the mathematical models mentioned earlier in regards to Classification can be applied to Cluster Analysis as well.

Prediction analysis is related to regression techniques. The key idea of prediction analysis is to discover the relationship between the dependent and independent variables, the relationship between the independent variables (one versus Another, one versus the rest, and so on). For example, if sales is an independent variable, then profit may be a dependent variable. By using historical data from both sales and profit, either linear or nonlinear regression techniques can produce a fitted regression curve that can be used for profit prediction in the future.

Sequential Pattern analysis seeks to find similar patterns in data transaction over a business period. These patterns can be used by business analysts to identify relationships among data. The mathematical models behind Sequential Patterns are logic rules, fuzzy logic, and so on. As an extension of Sequential Patterns, *Similar Time Sequences* are applied to discover sequences similar to a known sequence over both past and current business periods. In the data mining stage, several similar sequences can be studied to identify future trends in transaction development. This approach is useful in dealing with databases that have time-series characteristics.

Evaluation

The data interpretation stage is very critical. It assimilates knowledge from mined data. Two issues are essential. One is how to recognize the business value from knowledge patterns discovered in the data mining stage. Another issue is which visualization tool should be used to show the data mining results. Determining the business value from discovered knowledge patterns is similar to playing "puzzles." The mined data is a puzzle that needs to be put together for a business purpose. This operation depends on the interaction between data analysts, business analysts and decision makers (such as managers or CEOs). Because data analysts may not be fully aware of the purpose of the data mining goal or objective, and while business analysts may not understand the results of sophisticated mathematical solutions, interaction between them is necessary. In order to properly interpret knowledge patterns, it's important to choose an appropriate visualization tool. Many visualization packages and tools are available, including pie charts, histograms, box plots, scatter plots, and distributions. Good interpretation leads to productive business decisions, while poor interpretation analysis may miss useful information. Normally, the simpler the graphical interpretation, the easier it is for end users to understand.

Deployment

The results of the data mining study need to be reported back to project sponsors. The data mining study has uncovered new knowledge, which needs to be tied to the original data mining project goals. Management will then be in a position to apply this new understanding of their business environment.

It is important that the knowledge gained from a particular data mining study be monitored for change. Customer behavior changes over time, and what was true during the period when the data was collected may have already change. If fundamental changes occur, the knowledge uncovered is

no longer true. Therefore, it's critical that the domain of interest be monitored during its period of deployment.

SEMMA

In order to be applied successfully, the data mining solution must be viewed as a process rather than a set of tools or techniques. In addition to the CRISP-DM there is yet another well-known methodology developed by the SAS Institute, called SEMMA. The acronym SEMMA stands for *s*ample, *e*xplore, *m*odify, *m*odel, *a*ssess. Beginning with a statistically representative sample of your data, SEMMA intends to make it easy to apply exploratory statistical and visualization techniques, select and transform the most significant predictive variables, model the variables to predict outcomes, and finally confirm a model's accuracy. A pictorial representation of SEMMA is given in Fig. 2.2.

By assessing the outcome of each stage in the SEMMA process, one can determine how to model new questions raised by the previous results, and thus proceed back to the exploration phase for additional refinement of the data. That is, as is the case in CRISP-DM, SEMMA also driven by a highly iterative experimentation cycle.

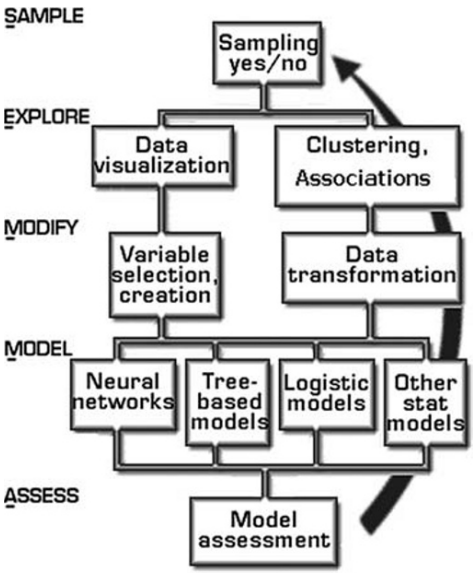

Fig. 2.2. Schematic of SEMMA (original from SAS Institute)

Steps in SEMMA Process

Step 1 (Sample): This is where a portion of a large data set (big enough to contain the significant information yet small enough to manipulate quickly) is extracted. For optimal cost and computational performance, some (including the SAS Institute) advocates a sampling strategy, which applies a reliable, statistically representative sample of the full detail data. In the case of very large datasets, mining a representative sample instead of the whole volume may drastically reduce the processing time required to get crucial business information. If general patterns appear in the data as a whole, these will be traceable in a representative sample. If a niche (a rare pattern) is so tiny that it is not represented in a sample and yet so important that it influences the big picture, it should be discovered using exploratory data description methods. It is also advised to create partitioned data sets for better accuracy assessment.

- Training – used for model fitting.
- Validation – used for assessment and to prevent over fitting.
- Test – used to obtain an honest assessment of how well a model generalizes.

Step 2 (Explore): This is where the user searched for unanticipated trends and anomalies in order to gain a better understanding of the data set. After sampling your data, the next step is to explore them visually or numerically for inherent trends or groupings. Exploration helps refine and redirect the discovery process. If visual exploration does not reveal clear trends, one can explore the data through statistical techniques including factor analysis, correspondence analysis, and clustering. For example, in data mining for a direct mail campaign, clustering might reveal groups of customers with distinct ordering patterns. Limiting the discovery process to each of these distinct groups individually may increase the likelihood of exploring richer patterns that may not be strong enough to be detected if the whole dataset is to be processed together.

Step 3 (Modify): This is where the user creates, selects, and transforms the variables upon which to focus the model construction process. Based on the discoveries in the exploration phase, one may need to manipulate data to include information such as the grouping of customers and significant subgroups, or to introduce new variables. It may also be necessary to look for outliers and reduce the number of variables, to narrow them down to the most significant ones. One may also need to modify data when the “mined”

data change. Because data mining is a dynamic, iterative process, you can update data mining methods or models when new information is available.

Step 4 (Model): This is where the user searches for a variable combination that reliably predicts a desired outcome. Once you prepare your data, you are ready to construct models that explain patterns in the data. Modeling techniques in data mining include artificial neural networks, decision trees, rough set analysis, support vector machines, logistic models, and other statistical models – such as time series analysis, memory-based reasoning, and principal component analysis. Each type of model has particular strengths, and is appropriate within specific data mining situations depending on the data. For example, artificial neural networks are very good at fitting highly complex nonlinear relationships while Rough sets analysis is know to produce reliable results with uncertain and imprecise problem situations.

Step 5 (Assess): This is where the user evaluates the usefulness and the reliability of findings from the data mining process. In this final step of the data mining process user assesses the models to estimate how well it performs. A common means of assessing a model is to apply it to a portion of data set put aside (and not used during the model building) during the sampling stage. If the model is valid, it should work for this reserved sample as well as for the sample used to construct the model. Similarly, you can test the model against known data. For example, if you know which customers in a file had high retention rates and your model predicts retention, you can check

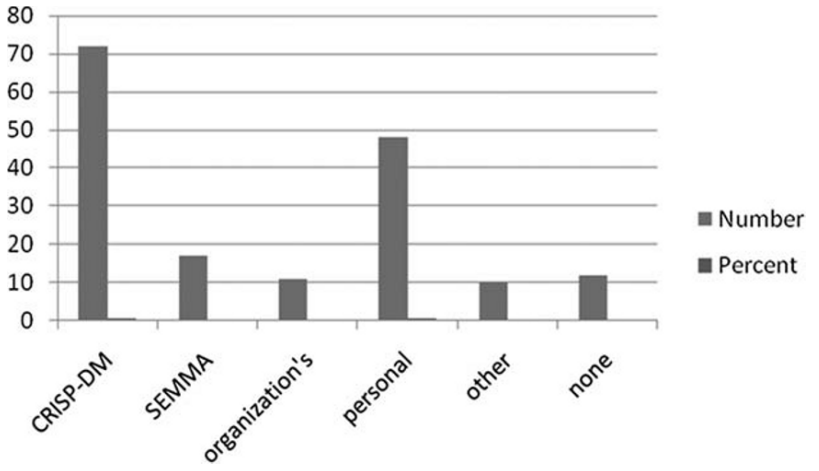

Fig. 2.3. Poll results – data mining methodology (conducted by KDNuggets.com on April 2004)

to see whether the model selects these customers accurately. In addition, practical applications of the model, such as partial mailings in a direct mail campaign, help prove its validity. The data mining web-site KDNuggets provided the data shown in Fig. 2.3 concerning relative use of data mining methodologies.

The SEMMA approach is completely compatible with the CRISP approach. Both aid the knowledge discovery process. Once models are obtained and tested, they can then be deployed to gain value with respect to business or research application.

Example Data Mining Process Application

Nayak and Qiu (2005) demonstrated the data mining process in an Australian software development project.[2] We will first relate their reported process, and then compare this with the CRISP and SEMMA frameworks.

The project owner was an international telecommunication company which undertook over 50 software projects annually. Processes were organized for Software Configuration Management, Software Risk Management, Software Project Metric Reporting, and Software Problem Report Management. Nayak and Qiu were interested in mining the

Table 2.1. Selected attributes from problem reports

Attribute	Description
Synopsis	Main issues
Responsibility	Individuals assigned
Confidentiality	Yes or no
Environment	Windows, Unix, etc.
Release note	Fixing comment
Audit trail	Process progress
Arrival date	
Close date	
Severity	Text describing the bug and impact on system
Priority	High, Medium, Low
State	Open, Active, Analysed, Suspended, Closed, Resolved, Feedback
Class	Sw-bug, Doc-bug, Change-request, Support, Mistaken, Duplicate

[2] R. Nayak, Tian Qiu (2005). A data mining application: Analysis of problems occurring during a software project development process, *International Journal of Software Engineering* 15:4, 647–663.

data from the Software Problem Reports. All problem reports were collected throughout the company (over 40,000 reports). For each report, data was available to include data shown in Table 2.1:

The data mining process reported included goal definition, data preprocessing, data modeling, and analysis of results.

1. Goal Definition

Data mining was expected to be useful in two areas. The first involved the early estimation and planning stage of a software project, company engineers have to estimate the number of lines of code, the kind of documents to be delivered, and estimated times. Accuracy at this stage would vastly improve project selection decisions. Little tool support was available for these activities, and estimates of these three attributes were based on experience supported by statistics on past projects. Thus projects involving new types of work were difficult to estimate with confidence. The second area of data mining application concerned the data collection system, which had limited information retrieval capability. Data was stored in flat files, and it was difficult to gather information related to specific issues.

2. Data Pre-Processing

This step consisted of attribute selection, data cleaning, and data transformation.

Data Field Selection: Some of the data was not pertinent to the data mining exercise, and was ignored. Of the variables given in Table 2.1, Confidentiality, Environment, Release note, and Audit trail were ignored as having no data mining value. They were, however, used during preprocessing and post-processing to aid in data selection and gaining better understanding of rules generated. For data stability, only problem reports for State values of Closed were selected.

Whenever a problem report was created, the project leader had to determine how long the fix took, how many people were involved, customer impact severity, impact on cost and schedule, and type of problem (software bug or design flaw). Thus the attributes listed below were selected as most important:

- Severity
- Priority
- Class

- Arrival-Date
- Close-Date
- Responsible
- Synopsis

The first five attributes had fixed values, and the Responsible attribute was converted to a count of those assigned to the problem. All of these attributes could be dealt with through conventional data mining tools. Synopsis was text data requiring text mining. Class was selected as the target attribute, with the possible outcomes given in Table 2.2:

Data Cleaning: Cleaning involved identification of missing, inconsistent, or mistaken values. Tools used in this process step included graphical tools to provide a picture of distributions, and statistics such as maxima, minima, mean values, and skew. Some entries were clearly invalid, caused by either human error or the evolution of the problem reporting system. For instance, over time, input for the Class attribute changed from SW-bug to sw-bug. Those errors that were correctable were corrected. If all errors detected for a report were not corrected, that report was discarded from the study.

Data Transformation: The attributes Arrival-Date and Close-Date were useful in this study to calculate the duration. Additional information was required, to include time zone. The Responsible attribute contained information identified how many people were involved. An attribute Time-to-fix was created multiplying the duration times the number of people, and then categorized into discrete values of 1 day, 3 days, 7 days, 14 days, 30 days, 90 days, 180 days, and 360 days (representing over one person-year).

In this application, 11,000 of the original 40,000 problem reports were left. They came from over 120 projects completed over the period 1996–2000. Four attributes were obtained:

Table 2.2. Class outcomes

Sw-bug	Bug from software code implementation
Doc-bug	Bug from documents directly related to the software product
Change-request	Customer enhancement request
Support	Bug from tools or documents, not the software product itself
Mistaken	Error in either software or document
Duplicate	Problem already covered in another problem report

- Time-to-fix
- Class
- Severity
- Priority

Text-mining was applied to 11,364 records, of which 364 had no time values so 11,000 were used for conventional data mining classification.

3. Data Modeling

Data mining provides functionality not provided by general database query techniques, which can't deal with the large number of records with high dimensional structures. Data mining provided useful functionality to answer questions such as the type of project documents requiring a great deal of development team time for bug repair, or the impact for various attribute values of synopsis, severity, priority, and class. A number of data mining tools were used.

- Prediction modeling was useful for evaluation of time consumption, giving sounder estimates for project estimation and planning.
- Link analysis was useful in discovering associations between attribute values.
- Text mining was useful in analyzing the Synopsis field.

Data mining software CBA was used for both classification and association rule analysis, C5 for classification, and TextAnalyst for text mining. An example classification rule was:

IF Severity non-critical AND Priority medium
THEN Class is Document with 70.72% confidence with support value of 6.5%

There were 352 problem reports in the training data set having these conditions, but only 256 satisfied the rule's conclusion.

Another rule including time-to-fix was more stringent:

IF 21 ≤ time-to-fix ≤ 108
AND Severity non-critical AND Priority medium
THEN Class is Document with 82.70% confidence with support value of 2.7%

There were 185 problem reports in the training data set with these conditions, 153 of which satisfied the rule's conclusion.

4. Analysis of Results

Classification and Association Rule Mining: Data was stratified using choice-based sampling rather than random sampling. This provided an equal number of samples for each target attribute field value. This improved the probability of obtaining rules for groups with small value counts (thus balancing the data). Three different training sets of varying size were generated. The first data set included 1,224 problem reports from one software project. The second data set consisted of equally distributed values from 3,400 problem reports selected from all software projects. The third data set consisted of 5,381 problem reports selected from all projects.

Minimum support and confidence were used to control rule modeling. Minimum support is a constraint requiring at least the stated number of cases be present in the training set. A high minimum support will yield fewer rules. Confidence is the strength of a rule as measured by the correct classification of cases. In practice, these are difficult to set ahead of analysis, and thus combinations of minimum support and confidence were used.

In this application, it was difficult for the CBA software to obtain correct classification on test data above 50%. The use of equal density of cases was not found to yield more accurate models in this study, although it appears a rational approach for further investigation. Using multiple support levels was also not found to improve error rates, and single support mining yielded a smaller number of rules. However, useful rules were obtained.

C5 was also applied for classification mining. C5 used cross validation, which splits the dataset into subsets (folds), treating each fold as a test case and the rest as training sets in hopes of finding a better result than a single training set process. C5 also has a boosting option, which generates and combines multiple classifiers in efforts to improve predictive accuracy. Here C5 yielded larger rule sets, with slightly better fits with training data, although at roughly the same level. Cross validation and boosting would not yield additional rules, but would focus on more accurate rules.

Text Mining: Pure text for the Synopsis attribute was categorized into a series of specific document types, such as “SRS – missing requirements” (with SRS standing for software requirement specification), “SRS – ability to turn off sending of SOH”, “Changes needed to SCMP_2.0.0” and so forth. TextAnalyst was used. This product builds a semantic network for text data investigation. Each element in the semantic network is assigned a

weight value, and relationships to other elements in the network, which are also assigned a weight value. Users are not required to specify predefined rules to build the semantic network. TextAnalyst provided a semantic network tree containing the most important words or word combinations (concepts), and reported relations and weights among these concepts ranging from 0 to 100, roughly analogous to probability. Text mining was applied to 11,226 cases.

Comparison of CRISP & SEMMA

The Nayak and Qiu case demonstrates a data mining process for a specific application, involving interesting aspects of data cleaning and transformation requirements, as well as a wide variety of data types, to include text. CRISP and SEMMA were created as broad frameworks, which need to be adapted to specific circumstances (see Table 2.3). We will now review how the Nayak and Qiu case fits these frameworks.

Nayak and Qiu started off with a clearly stated set of goals – to develop tools that would better utilize the wealth of data in software project problem reports.

They examined data available, and identified what would be useful. Much of the information from the problem reports was discarded. SEMMA includes sampling efforts here, which CRISP would include in data preparation, and which Nayak and Qiu accomplished after data transformation. Training and test sets were used as part of the software application.

Table 2.3. Comparison of methods

CRISP	SEMMA	Nayak & Qiu
Business understanding	Assumes well-defined question	Goals were defined Develop tools to better utilize problem reports
Data understanding	*S*ample *E*xplore	Looked at data in problem reports
Data preparation	*M*odify data	Data pre-processing Data cleaning Data transformation
Modeling	*M*odel	Data modeling
Evaluation	*A*ssess	Analyzing results
Deployment		

Data was cleaned, and reports with missing observations were discarded from the study. Data preparation involved data transformation. Specifically, they used two problem report attributes to generate project duration, which was further transformed by multiplying by the number of people assigned (available by name, but only counts were needed). The resultant measure of effort was further transformed into categories that reflected relative importance without cluttering detail.

Modeling included classification and association rule analysis from the first software tool (CBA), a replication of classification with C5, and independent text analysis with TextAnalyst. Nayak and Qiu generated a variety of models by manipulating minimum support and confidence levels in the software.

Evaluation (assessment) was accomplished by Nayak and Qiu through analysis of results in terms of the number of rules, as well as accuracy of classification models as applied to the test set.

CRISP addresses the deployment of data mining models, which is implicit in any study. Nayak and Qiu's models were presumably deployed, but that was not addressed in their report.

Handling Data

A recent data mining study in insurance applied a knowledge discovery process.[3] This process involved iteratively applying the steps that we covered in CRISP-DM, and demonstrating how the methodology can work in practice.

Stage 1. Business Understanding

A model was needed to predict which customers would be insolvent early enough for the firm to take preventive measures (or measures to avert losing good customers). This goal included minimizing the misclassification of legitimate customers.

In this case, the billing period was 2 months. Customers used their phone for 4 weeks, and received bills about 1 week later. Payment was due a month after the date of billing. In the industry, companies typically gave customers about 2 weeks after the due-date before taking action, at which time the phone was disconnected if the unpaid bill was greater than a set

[3] S. Daskalaki, I. Kopanas, M. Goudara, N. Avouris (2003). Data mining for decision support on customer insolvency in the telecommunications business, *European Journal of Operational Research* 145, 239–255.

amount. Bills were sent every month for another 6 months, during which period the late customer could make payment arrangements. If no payment was received at the end of this 6-month period, the unpaid balance was transferred to the uncollectible category.

This study hypothesized that insolvent customers would change their calling habits and phone usage during a critical period before and immediately after termination of the billing period. Changes in calling habits, combined with paying patterns were tested for their ability to provide sound predictions of future insolvencies.

Stage 2. Data Understanding

Static customer information was available from customer files. Time-dependent data was available on bills, payments, and usage. Data came from several databases, but all of these databases were internal to the company. A data warehouse was built to gather and organize this data. The data was coded to protect customer privacy. Data included customer information, phone usage from switching centers, billing information, payment reports by customer, phone disconnections due to a failure to pay, phone reconnections after payment, and reports of permanent contract nullifications.

Data was selected for 100,000 customers covering a 17-month period, and was collected from one rural/agricultural region of customers, a semi-rural touring area, and an urban/industrial area in order to assure representative cross-sections of the company's customer base. The data warehouse used over 10 gigabytes of storage for raw data.

Stage 3. Data Preparation

The data was tested for quality, and data that wasn't useful for the study was filtered out. Heterogeneous data items were interrelated. As examples, it was clear that inexpensive calls had little impact on the study. This allowed a 50% reduction in the total volume of data. The low percentage of fraudulent cases made it necessary to clean the data from missing or erroneous values due to different recording practices within the organization and the dispersion of data sources. Thus it was necessary to cross-check data such as phone disconnections. The lagged data required synchronization of different data elements.

Data synchronization revealed a number of insolvent customers with missing information that had to be deleted from the data set. It was thus necessary to reduce and project data, so information was grouped by account to make data manipulation easier, and customer data was aggregated

by 2-week periods. Statistics were applied to find characteristics that were discriminant factors for solvent versus insolvent customers. Data included the following:

- Telephone account category (23 categories, such as payphone, business, and so on).
- Average amount owed was calculated for all solvent and insolvent customers. Insolvent customers had significantly higher averages across all categories of account.
- Extra charges on bills were identified by comparing total charges for phone usage for the period as opposed to balances carried forward or purchases of hardware or other services. This also proved to be statistically significant across the two outcome categories.
- Payment by installments was investigated. However, this variable was not found to be statistically significant.

Stage 4. Modeling

The prediction problem was classification, with two classes: most possibly solvent (99.3% of the cases) and most possibly insolvent (0.7% of the cases). Thus, the count of insolvent cases was very small in a given billing period. The costs of error varied widely in the two categories. This has been noted by many as a very difficult classification problem.

A new dataset was created through stratified sampling for solvent customers, altering the distribution of customers to be 90% solvent and 10% insolvent. All of the insolvent cases were retained, while care was taken to maintain a proportional representation of the solvent set of data over variables such as geographical region. A dataset of 2,066 total cases was developed.

A critical period for each phone account was established. For those accounts that were nullified, this critical period was the last 15 two-week periods prior to service interruption. For accounts that remained active, the critical period was set as a similar period to possible disruption. There were six possible disruption dates per year. For the active accounts, one of these six dates was selected at random.

For each account, variables were defined by counting the appropriate measure for every 2-week period in the critical period for that observation. At the end of this phase, new variables were created to describe phone usage by account compared to a moving average of four previous 2-week periods. At this stage, there were 46 variables as candidate discriminating factors. These variables included 40 variables measured as call habits over

15 two-week periods, as well as variables concerning the type of customer, whether or not a customer was new, and four variables relating to customer bill payment.

Discriminant analysis, decision trees and neural network algorithms were used to test hypotheses over the reduced data set of 2,066 cases measured over 46 variables. Discriminant analysis yielded a linear model, the neural network came out as a nonlinear model, and the decision tree was a rule-based classifier.

Stage 5. Evaluation

Experiments were conducted to test and compare performance. The data set was divided into a training set (about two-thirds of the 2,066 cases) and test set (the remaining cases). Classification errors are commonly displayed in *coincidence matrices* (called confusion matrices by some). A coincidence matrix shows the count of cases correctly classified, as well as the count of cases classified in each incorrect category. But in many data mining studies, the model may be very good at classifying one category, while very poor at classifying another category. The primary value of the coincidence matrix is that it identifies what kinds of errors are made. It may be much more important to avoid one kind of error than another. For instance, a bank loan officer suffers a great deal more from giving a loan to someone who's expected to repay and does not than making the mistake of not giving a loan to an applicant who actually would have paid. Both instances would be classification errors, but in data mining, often one category of error is much more important than another. Coincidence matrices provide a means of focusing on what kinds of errors particular models tend to make.

A way to reflect relative error importance is through cost. This is a relatively simple idea, allowing the user to assign relative costs by type of error. For instance, if our model predicted that an account was insolvent, that might involve an average write-off of $200. On the other hand, waiting for an account that ultimately was repaid might involve a cost of $10. Thus, there would be a major difference in the cost of errors in this case. Treating a case that turned out to be repaid as a dead account would risk the loss of $190, in addition to alienating the customer (which may or may not have future profitability implications). Conversely, treating an account that was never going to be repaid may involve carrying the account on the books longer than needed, at an additional cost of $10. Here, a cost function for the coincidence matrix could be:

$$\$190 \times (\text{closing good account}) + \$10 \times (\text{keeping bad account open})$$

(Note that we used our own dollar costs for purposes of demonstration, and these were not based on the real case.) This measure (like the correct classification rate) can be used to compare alternative models.

SPSS was used for discriminant analysis, including a stepwise forward selection procedure. The best model included 17 of the available 46 variables. Using equal misclassification costs yielded the coincidence matrix shown in Table 2.4.

Overall classification accuracy is obtained by dividing the correct number of classifications (50 + 578 = 628) by the total number of cases (718). Thus, the test data was correctly classified in 87.5. The cost function value here was:

$$\$190 \times 76 + \$10 \times 14 = \$14{,}580$$

The high proportion of actually solvent cases classified as insolvent was judged to be unacceptable, because it would chase away too many good customers. The experiment was reconducted using a-priori probabilities. This improved output significantly, as shown in the coincidence matrix in Table 2.5.

The test data was correctly classified in 93.0% of the cases. For the training data, this figure was 93.6%. Models usually fit training data a little better than test data, but that's because they were built on training data. Independent test data provides a much better test. The accuracy for insolvent customers, which is very important because it costs so much more, decreased from 78% in the training data to 56% in the test data. The cost function value here was the following:

$$\$190 \times 22 + \$10 \times 28 = \$4{,}460$$

Table 2.4. Coincidence matrix – equal misclassification costs

Telephone bill	Model insolvent	Model solvent	
Actual Insolvent	50	14	64
Actual Solvent	76	578	654
	126	593	718

Table 2.5. Coincidence matrix – unequal misclassification costs

Telephone bill	Model insolvent	Model solvent	
Actual Insolvent	36	28	64
Actual Solvent	22	632	654
	58	660	718

From a total cost perspective, the model utilizing unequal misclassification costs (using real costs) was considered more useful.

The 17 variables identified in the discriminant analysis were used for the other two models. The same training and test sets were employed. The training set was used to build a rule-based classifier model. The coincidence matrix for the test set is shown in Table 2.6.

Thus the test data was correctly classified in 95.26% of the cases. For the training data, this figure was 95.3%. The cost function value here was

$$\$190 \times 8 + \$10 \times 26 = \$1{,}780$$

This was an improvement over the discriminant analysis model.

A number of experiments were conducted with a neural network model using the same 17 variables and training set. The resulting coincidence matrix over the test data is shown in Table 2.7.

The test data was correctly classified in 92.9% of the cases. For the training data, this figure was 94.1%. The cost function value here was

$$\$190 \times 11 + \$10 \times 40 = \$2{,}490$$

However, these results were inferior to that of the decision tree model.

The first objective was to maximize accuracy of predicting insolvent customers. The decision tree classifier appeared to be best at doing that. The second objective was to minimize the error rate for solvent customers. The neural network model was close to the performance of the decision tree model. It was decided to use all three models on a case-by-case basis.

Table 2.6. Coincidence matrix – the rule-based model

Telephone bill	Model insolvent	Model solvent	
Actual Insolvent	38	26	64
Actual Solvent	8	646	654
	46	672	718

Table 2.7. Coincidence matrix – the neural network model

Telephone bill	Model insolvent	Model solvent	
Actual Insolvent	24	40	64
Actual Solvent	11	643	654
	35	683	718

Table 2.8. Coincidence matrix – combined models

Telephone bill	Model insolvent	Model solvent	Unclassified	Total
Actual Insolvent	19	17	28	64
Actual Solvent	1	626	27	654
	20	643	91	718

Stage 6. Deployment

Every customer was examined using all three algorithms. If all three agreed on classification, that result was adopted. If there was disagreement in the model results, the customer was categorized as unclassified. Using this scheme over the test set yielded the coincidence matrix shown in Table 2.8.

Thus, the test data was correctly classified in 89.8% of the cases. But only one actually solvent customer would have been disconnected without further analysis. The cost function value here was

$$\$190 \times 1 + \$10 \times 17 = \$360$$

The steps used in this application match the six stages we have presented. Data Selection relates to Learning the application domain and Creating a target dataset. Data Preprocessing involves Data Cleaning and preprocessing. Data Transformation involves Data Reduction and projection. Data Mining was expanded in the earlier application to include (1) choosing the function of data mining, (2) choosing the data mining algorithms, and (3) data mining. Data Interpretation involves the interpretation and use of discovered knowledge.

Summary

The industry-standard CRISP-DM data mining process has six stages: (1) Business Understanding, (2) Data Understanding, (3) Data Preparation, (4) Modeling, (5) Evaluation, and (6) Deployment. Data selection and understanding, preparation, and model interpretation require teamwork between data mining analysts and business analysts, while data transformation and data mining are conducted by data mining analysts alone. Each stage is a preparation for the next stage. In the remaining

chapters of this book, we'll discuss details regarding this process from a different perspective, such as data mining tools and applications. This will provide the reader with a better understanding as to why the correct process sometimes is even more important than the correct performance of the methodology.