CHAPTER

1

Overview of Predictive Analytics

A small direct response company had developed dozens of programs in cooperation with major brands to sell books and DVDs. These affinity programs were very successful, but required considerable up-front work to develop the creative content and determine which customers, already engaged with the brand, were worth the significant marketing spend to purchase the books or DVDs on subscription. Typically, they first developed test mailings on a moderately sized sample to determine if the expected response rates were high enough to justify a larger program.

One analyst with the company identified a way to help the company become more profitable. What if one could identify the key characteristics of those who responded to the test mailing? Furthermore, what if one could generate a score for these customers and determine what minimum score would result in a high enough response rate to make the campaign profitable? The analyst discovered predictive analytics techniques that could be used for both purposes, finding key customer characteristics and using those characteristics to generate a score that could be used to determine which customers to mail.

Two decades before, the owner of a small company in Virginia had a compelling idea: Improve the accuracy and flexibility of guided munitions using optimal control. The owner and president, Roger Barron, began the process of deriving the complex mathematics behind optimal control using a technique known as variational calculus and hired a graduate student to assist him in the task. Programmers then implemented the mathematics in computer code so they could simulate thousands of scenarios. For each trajectory, the variational calculus minimized the miss distance while maximizing speed at impact as well as the angle of impact.

The variational calculus algorithm succeeded in identifying the optimal sequence of commands: how much the fins (control surfaces) needed to change the path of the munition to follow the optimal path to the target. The concept worked in simulation in the thousands of optimal trajectories that were run. Moreover, the mathematics worked on several munitions, one of which was the MK82 glide bomb, fitted (in simulation) with an inertial guidance unit to control the fins: an early smart-bomb.

There was a problem, however. The variational calculus was so computationally complex that the small computers on-board could not solve the problem in real time. But what if one could *estimate* the optimal guidance commands at any time during the flight from observable characteristics of the flight? After all, the guidance unit can compute where the bomb is in space, how fast it is going, and the distance of the target that was programmed into the unit when it was launched. If the estimates of the optimum guidance commands were close enough to the actual optimal path, it would be *near optimal* and still succeed. Predictive models were built to do exactly this. The system was called *Optimal Path-to-Go* guidance.

These two programs designed by two different companies seemingly could not be more different. One program knows characteristics of people, such as demographics and their level of engagement with a brand, and tries to predict a human decision. The second program knows locations of a bomb in space and tries to predict the best physical action for it to hit a target.

But they share something in common: They both need to estimate values that are unknown but tremendously useful. For the affinity programs, the models estimate whether or not an individual will respond to a campaign, and for the guidance program, the models estimate the best guidance command. In this sense, these two programs are very similar because they both involve predicting a value or values that are known historically, but are unknown at the time a decision is needed. Not only are these programs related in this sense, but they are far from unique; there are countless decisions businesses and government agencies make every day that can be improved by using historic data as an aid to making decisions or even to automate the decisions themselves.

This book describes the back-story behind how analysts build the predictive models like the ones described in these two programs. There is science behind much of what predictive modelers do, yet there is also plenty of *art*, where no theory can inform us as to the best action, but experience provides principles by which tradeoffs can be made as solutions are found. Without the art, the science would only be able to solve a small subset of problems we face. Without the science, we would be like a plane without a rudder or a kite without a tail, moving at a rapid pace without any control, unable to achieve our objectives.

What Is Analytics?

Analytics is the process of using computational methods to discover and report influential patterns in data. The goal of analytics is to gain insight and often to affect decisions. Data is necessarily a measure of historic information so, by definition, analytics examines historic data. The term itself rose to prominence in 2005, in large part due to the introduction of Google Analytics. Nevertheless, the ideas behind analytics are not new at all but have been represented by different terms throughout the decades, including *cybernetics*, *data analysis*, *neural networks*, *pattern recognition*, *statistics*, *knowledge discovery*, data *mining*, and now even *data science*.

The rise of analytics in recent years is pragmatic: As organizations collect more data and begin to summarize it, there is a natural progression toward using the data to improve estimates, forecasts, decisions, and ultimately, efficiency.

What Is Predictive Analytics?

Predictive analytics is the process of discovering interesting and meaningful patterns in data. It draws from several related disciplines, some of which have been used to discover patterns in data for more than 100 years, including pattern recognition, statistics, machine learning, artificial intelligence, and data mining. What differentiates predictive analytics from other types of analytics?

First, predictive analytics is data-driven, meaning that algorithms derive key characteristic of the models from the data itself rather than from assumptions made by the analyst. Put another way, data-driven algorithms *induce* models from the data. The induction process can include identification of variables to be included in the model, parameters that define the model, weights or coefficients in the model, or model complexity.

Second, predictive analytics algorithms automate the process of finding the patterns from the data. Powerful induction algorithms not only discover coefficients or weights for the models, but also the very form of the models. Decision trees algorithms, for example, learn which of the candidate inputs best predict a target variable in addition to identifying which values of the variables to use in building predictions. Other algorithms can be modified to perform searches, using exhaustive or greedy searches to find the best set of inputs and model parameters. If the variable helps reduce model error, the variable is included in the model. Otherwise, if the variable does not help to reduce model error, it is eliminated.

Another automation task available in many software packages and algorithms automates the process of transforming input variables so that they can be used effectively in the predictive models. For example, if there are a hundred variables that are candidate inputs to models that can be or should be transformed to remove skew, you can do this with some predictive analytics software in a single step rather than programming all one hundred transformations one at a time.

Predictive analytics doesn't do anything that any analyst couldn't accomplish with pencil and paper or a spreadsheet if given enough time; the algorithms, while powerful, have no common sense. Consider a supervised learning data set with 50 inputs and a single binary target variable with values 0 and 1. One way to try to identify which of the inputs is most related to the target variable is to plot each variable, one at a time, in a histogram. The target variable can be superimposed on the histogram, as shown in Figure 1-1. With 50 inputs, you need to look at 50 histograms. This is not uncommon for predictive modelers to do.

If the patterns require examining two variables at a time, you can do so with a scatter plot. For 50 variables, there are 1,225 possible scatter plots to examine. A dedicated predictive modeler might actually do this, although it will take some time. However, if the patterns require that you examine three variables simultaneously, you would need to examine 19,600 3D scatter plots in order to examine all the possible three-way combinations. Even the most dedicated modelers will be hard-pressed to spend the time needed to examine so many plots.

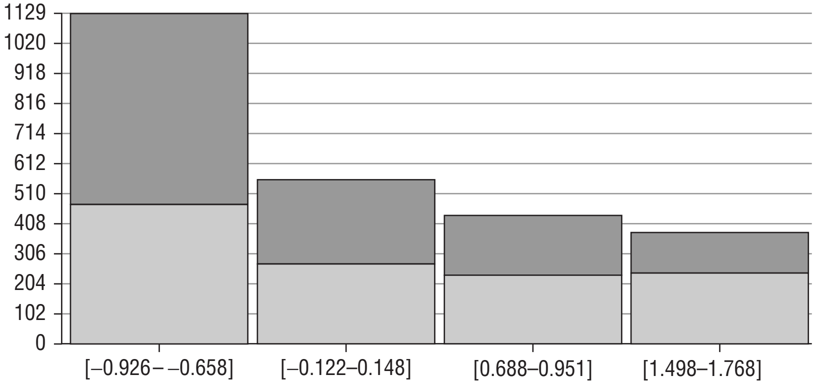

Figure 1-1: Histogram

You need algorithms to sift through all of the potential combinations of inputs in the data—the patterns—and identify which ones are the most interesting. The analyst can then focus on these patterns, undoubtedly a much smaller number of inputs to examine. Of the 19,600 three-way combinations of inputs, it may be that a predictive model identifies six of the variables as the most significant contributors to accurate models. In addition, of these six variables, the top three are particularly good predictors and much better than any two variables by themselves. Now you have a manageable subset of plots to consider: 63 instead of nearly 20,000. This is one of the most powerful aspects of predictive analytics: identifying which inputs are the most important contributors to patterns in the data.

Supervised vs. Unsupervised Learning

Algorithms for predictive modeling are often divided into two groups: supervised learning methods and unsupervised learning methods. In supervised learning models, the *supervisor* is the target variable, a column in the data representing values to predict from other columns in the data. The target variable is chosen to represent the answer to a question the organization would like to answer or a value unknown at the time the model is used that would help in decisions. Sometimes supervised learning is also called *predictive modeling*. The primary predictive modeling algorithms are *classification* for categorical target variables or *regression* for continuous target variables.

Examples of target variables include whether a customer purchased a product, the amount of a purchase, if a transaction was fraudulent, if a customer stated they enjoyed a movie, how many days will transpire before the next gift a donor will make, if a loan defaulted, and if a product failed. Records without a value for the target variable cannot be used in building predictive models.

Unsupervised learning, sometimes called *descriptive modeling*, has no target variable. The inputs are analyzed and grouped or clustered based on the proximity of input values to one another. Each group or cluster is given a label to indicate which group a record belongs to. In some applications, such as in customer analytics, unsupervised learning is just called segmentation because of the function of the models (segmenting customers into groups).

The key to supervised learning is that the inputs to the model are known but there are circumstances where the target variable is unobserved or unknown. The most common reason for this is a target variable that is an event, decision, or other behavior that takes place at a time future to the observed inputs to the model. Response models, cross-sell, and up-sell models work this way: Given what is known now about a customer, can you predict if they will purchase a particular product in the future?

Some definitions of *predictive analytics* emphasize the function of algorithms as forecasting or predicting *future* events or behavior. While this is often the case, it certainly isn't always the case. The target variable could represent an unobserved variable like a missing value. If a taxpayer didn't file a return in a prior year, predictive models can predict that missing value from other examples of tax returns where the values are known.

Parametric vs. Non-Parametric Models

Algorithms for predictive analytics include both parametric and non-parametric algorithms. Parametric algorithms (or models) assume known distributions in the data. Many parametric algorithms and statistical tests, although not all, assume normal distributions and find linear relationships in the data. Machine learning algorithms typically do not assume distributions and therefore are considered non-parametric or distribution-free models.

The advantage of parametric models is that if the distributions are known, extensive properties of the data are also known and therefore algorithms can be proven to have very specific properties related to errors, convergence, and certainty of learned coefficients. Because of the assumptions, however, the analyst often spends considerable time transforming the data so that these advantages can be realized.

Non-parametric models are far more flexible because they do not have underlying assumptions about the distribution of the data, saving the analyst considerable time in preparing data. However, far less is known about the data *a priori*, and therefore non-parametric algorithms are typically iterative, without any guarantee that the best or optimal solution has been found.

Business Intelligence

Business intelligence is a vast field of study that is the subject of entire books; this treatment is brief and intended to summarize the primary characteristics of business intelligence as they relate to predictive analytics. The output of many business intelligence analyses are reports or dashboards that summarize interesting characteristics of the data, often described as Key Performance Indicators (KPIs). The KPI reports are user-driven, determined by an analyst or decision-maker to represent a key descriptor to be used by the business. These reports can contain simple summaries or very complex, multidimensional measures. Interestingly, KPI is almost never used to describe measures of interest in predictive analytics software and conferences.

Typical business intelligence output is a report to be used by analysts and decision-makers. The following are typical questions that might be answered by business intelligence for fraud detection and customer analytics:

Fraud Detection

- How many cases were investigated last month?
- What was the success rate in collecting debts?
- How much revenue was recovered through collections?
- What was the ROI for the various collection avenues: letters, calls, agents?

- What was the close rate of cases in the past month? Past quarter? Past year?
- For debts that were closed out, how many days did it take on average to close out debts?
- For debts that were closed out, how many contacts with the debtor did it take to close out debt?

Customer Analytics

- What were the e-mail open, click-through, and response rates?
- Which regions/states/ZIPs had the highest response rates?
- Which products had the highest/lowest click-through rates?
- How many repeat purchasers were there last month?
- How many new subscriptions to the loyalty program were there?
- What is the average spend of those who belong to the loyalty program? Those who aren't a part of the loyalty program? Is this a significant difference?
- How many visits to the store/website did a person have?

These questions describe characteristics of the unit of analysis: a customer, a transaction, a product, a day, or even a ZIP code. Descriptions of the unit of analysis are contained in the columns of the data: the attributes. For fraud detection, the unit of analysis is sometimes a debt to be collected, or more generally a case. For customer analytics, the unit of analysis is frequently a customer but could be a visit (a single customer could visit many times and therefore will appear in the data many times).

Note that often these questions compare directly one attribute of interest with an outcome of interest. These questions were developed by a domain expert (whether an analyst, program manager, or other subject matter expert) as a way to describe interesting relationships in the data relevant to the company. In other words, these measures are user-driven.

Are these KPIs and reports actionable decisions in and of themselves? The answer is no, although they can be with small modifications. In the form of the report, you know what happened and can even identify why it happened in some cases. It isn't a great leap, however, to take reports and turn them into predictions. For example, a report that summarizes the response rates for each ZIP code can then use ZIP as a predictor of response rate.

If you consider the reports related to a target variable such as response rate, the equivalent machine learning approach is building a *decision stump*, a single condition rule that predicts the outcome. But this is a very simple way of approaching prediction.

Predictive Analytics vs. Business Intelligence

What if you reconstruct the two lists of questions in a different way, one that is focused more directly on decisions? From a predictive analytics perspective, you may find these questions are the ones asked.

Fraud Detection

- What is the likelihood that the transaction is fraudulent?
- What is the likelihood the invoice is fraudulent or warrants further investigation?
- Which characteristics of the transaction are most related to or most predictive of fraud (single characteristics and interactions)?
- What is the expected amount of fraud?
- What is the likelihood that a tax return is non-compliant?
- Which line items on a tax return contribute the most to the fraud score?
- Historically, which demographic and historic purchase patterns were most related to fraud?

Customer Analytics for Predictive Analytics

- What is the likelihood an e-mail will be opened?
- What is the likelihood a customer will click-through a link in an e-mail?
- Which product is a customer most likely to purchase if given the choice?
- How many e-mails should the customer receive to maximize the likelihood of a purchase?
- What is the best product to up-sell to the customer after they purchase a product?
- What is the visit volume expected on the website next week?
- What is the likelihood a product will sell out if it is put on sale?
- What is the estimated customer lifetime value (CLV) of each customer?

Notice the differences in the kinds of questions predictive analytics asks compared to business intelligence. The word "likelihood" appears often, meaning we are computing a probability that the pattern exists for a unit of analysis. In customer analytics, this could mean computing a probability that a customer is likely to purchase a product.

Implicit in the wording is that the measures require an examination of the groups of records comprising the unit of analysis. If the likelihood an individual customer will purchase a product is one percent, this means that for every 100 customers with the same pattern of measured attributes for this customer, one customer purchased the product in the historic data used to compute the likelihood. The comparable measure in the business intelligence lists would be described as a *rate* or a *percentage*; what is the response rate of customers with a particular purchase pattern.

The difference between the business intelligence and predictive analytics measures is that the business intelligence variables identified in the questions were, as already described, user driven. In the predictive analytics approach, the predictive modeling algorithms considered many patterns, sometimes all possible patterns, and determined which ones were most predictive of the measure of interest (likelihood). The discovery of the patterns is data driven.

This is also why many of the questions begin with the word "which." Asking *which* line items on a tax return are most related to noncompliance requires comparisons of the line items as they relate to noncompliance.

Do Predictive Models Just State the Obvious?

Often when presenting models to decision-makers, modelers may hear a familiar refrain: "I didn't need a model to tell me that!" But predictive models do more than just identify attributes that are related to a target variable. They identify the *best way* to predict the target. Of all the possible alternatives, all of the attributes that could predict the target and all of the interactions between the attributes, which combinations do the best job? The decision-maker may have been able to guess (hypothesize) that length or residence is a good attribute to predict a responder to a Medicare customer acquisition campaign, but that same person may not have known that the number of contacts is even more predictive, especially when the prospect has been mailed two to six times. Predictive models identify not only which variables are predictive, but how well they predict the target. Moreover, they also reveal which combinations are not just predictive of the target, but *how well* the combinations predict the target and how much better they predict than individual attributes do on their own.

Similarities between Business Intelligence and Predictive Analytics

Often, descriptions of the differences between business intelligence and predictive analytics stress that business intelligence is retrospective analysis, looking back into the past, whereas predictive analytics or prospective analysis predict future behavior. The "predicting the future" label is applied often to predictive analytics in general and the very questions described already imply this is the case. Questions such as "What is the likelihood a customer will purchase . . ." are forecasting future behavior.

Figure 1-2 shows a timeline relating data used to build predictive models or business intelligence reports. The vertical line in the middle is the time the model is being built (today). The data used to build the models is always to the left: historic data. When predictive models are built to predict a "future" event, the data selected to build the predictive models is rolled back to a time prior to the date the future event is known.

For example, if you are building models to predict whether a customer will respond to an e-mail campaign, you begin with the date the campaign cured (when all the responses have come in) to identify everyone who responded. This is the date for the label "target variable computed based on this date" in the figure. The attributes used as inputs must be known prior to the date of the mailing itself, so these values are collected to the left of the target variable collection date. In other words, the data is set up with all the modeling data in the past, but the target variable is still future to the date the attributes are collected in the timeline of the data used for modeling.

However, it's important to be clear that both business intelligence and predictive analytics analyses are built from the same data, and the data is historic in both cases. The assumption is that future behavior to the right of the vertical line in Figure 1-2 will be consistent with past behavior. If a predictive model identifies patterns in the past that predicted (in the past) that a customer would purchase a product, you assume this relationship will continue to be present in the future.

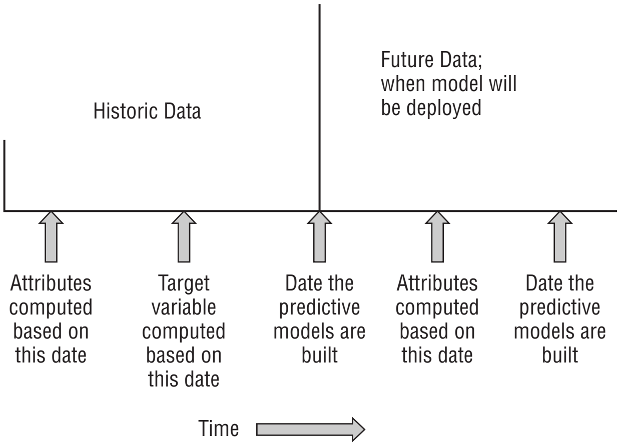
Historic Data
Future Data; when model will be deployed
Attributes computed based on this date
Target variable computed based on this date
Date the predictive models are built
Attributes computed based on this date
Date the predictive models are built
Time

Figure 1-2: Timeline for building predictive models

Predictive Analytics vs. Statistics

Predictive analytics and statistics have considerable overlap, with some statisticians arguing that predictive analytics is, at its core, an extension of statistics. Predictive modelers, for their part, often use algorithms and tests common in statistics as a part of their regular suite of techniques, sometimes without applying the diagnostics most statisticians would apply to ensure the models are built properly.

Since predictive analytics draws heavily from statistics, the field has taken to heart the amusing quote from statistician and creator of the bootstrap, Brad Efron: "Those who ignore Statistics are condemned to reinvent it." Nevertheless, there are significant differences between typical approaches of the two fields. Table 1-1 provides a short list of items that differ between the fields. Statistics is driven by theory in a way that predictive analytics is not, where many algorithms are drawn from other fields such as machine learning and artificial intelligence that have no provable optimum solution.

But perhaps the most fundamental difference between the fields is summarized in the last row of the table: For statistics, the model is king, whereas for predictive analytics, data is king.

Table 1-1: Statistics vs. Predictive Analytics

STATISTICS	PREDICTIVE ANALYTICS
Models based on theory: There is an optimum.	Models often based on non-parametric algorithms; no guaranteed optimum
Models typically linear.	Models typically nonlinear
Data typically smaller; algorithms often geared toward accuracy with small data	Scales to big data; algorithms not as efficient or stable for small data
The model is king.	Data is king.

Statistics and Analytics

In spite of the similarities between statistics and analytics, there is a difference in mindset that results in differences in how analyses are conducted. Statistics is often used to perform confirmatory analysis where a hypothesis about a relationship between inputs and an output is made, and the purpose of the analysis is to confirm or deny the relationship and quantify the degree of that confirmation or denial. Many analyses are highly structured, such as determining if a drug is effective in reducing the incidence of a particular disease.

Controls are essential to ensure that bias is not introduced into the model, thus misleading the analyst's interpretation of the model. Coefficients of models are critically important in understanding what the data are saying, and therefore great care is taken to transform the model inputs and outputs so they comply with assumptions of the modeling algorithms. If the study is predicting the effect of caloric intake, smoking, age, height, amount of exercise, and metabolism on an individual's weight, and one is to trust the relative contribution of each factor on an individual's weight, it is important to remove any bias due to the data itself so that the conclusions reflect the intent of the model. Bias in the data could result in misleading the analyst that the inputs to the model have more or less influence that they actually have, simply because of numeric problems in the data.

Residuals are also carefully examined to identify departure from a Normal distribution, although the requirement of normality lessens as the size of the data increases. If residuals are not random with constant variance, the statistician will modify the inputs and outputs until these problems are corrected.

Predictive Analytics and Statistics Contrasted

Predictive modelers, on the other hand, often show little concern for final parameters in the models except in very general terms. The key is often the predictive accuracy of the model and therefore the ability of the model to make and influence decisions. In contrast to the structured problem being solved through confirmatory analysis using statistics, predictive analytics often attempts to solve less structured business problems using data that was not even collected for the purpose of building models; it just happened to be around. Controls are often not in place in the data and therefore causality, very difficult to uncover even in structured problems, becomes exceedingly difficult to identify. Consider, for example, how you would go about identifying which marketing campaign to apply to a current customer for a digital retailer. This customer could receive content from any one of ten programs the e-mail marketing group has identified. The modeling data includes customers, their demographics, their prior behavior on the website and with e-mail they had received in the past, and their reaction to sample content from one of the ten programs. The reaction could be that they ignored the e-mail, opened it, clicked through the link, and ultimately purchased the product promoted in the e-mail. Predictive models can certainly be built to identify the best program of the ten to put into the e-mail based on a customer's behavior and demographics.

However, this is far from a controlled study. While this program is going on, each customer continues to interact with the website, seeing other promotions. The customer may have seen other display ads or conducted Google searches further influencing his or her behavior. The purpose of this kind of model cannot be to uncover fully why the customer behaves in a particular way because there are far too many unobserved, confounding influences. But that doesn't mean the model isn't useful.

Predictive modelers frequently approach problems in this more unstructured, even casual manner. The data, in whatever form it is found, drives the models. This isn't a problem as long as the data continues to be collected in a manner consistent with the data as it was used in the models; consistency in the data will increase the likelihood that there will be consistency in the model's predictions, and therefore how well the model affects decisions.

Predictive Analytics vs. Data Mining

Predictive analytics has much in common with its immediate predecessor, data mining; the algorithms and approaches are generally the same. Data mining has a history of applications in a wide variety of fields, including finance, engineering, manufacturing, biotechnology, customer relationship management, and marketing. I have treated the two fields as generally synonymous since "predictive analytics" became a popular term.

This general overlap between the two fields is further emphasized by how software vendors brand their products, using both data mining and predictive analytics (some emphasizing one term more than the other).

On the other hand, data mining has been caught up in the specter of privacy concerns, spam, malware, and unscrupulous marketers. In the early 2000s, congressional legislation was introduced several times to curtail specifically any data mining programs in the Department of Defense (DoD). Complaints were even waged against the use of data mining by the NSA, including a letter sent by Senator Russ Feingold to the National Security Agency (NSA) Director in 2006:

> *One element of the NSA's domestic spying program that has gotten too little attention is the government's reportedly widespread use of data mining technology to analyze the communications of ordinary Americans. Today I am calling on the Director of National Intelligence, the Defense Secretary and the Director of the NSA to explain whether and how the government is using data mining technology, and what authority it claims for doing so.*

In an interesting *déjà vu*, in 2013, information about NSA programs that sift through phone records was leaked to the media. As in 2006, concerns about privacy were again raised, but this time the mathematics behind the program, while typically described as data mining in the past, was now often described as predictive analytics.

Graduate programs in analytics often use both data mining and predictive analytics in their descriptions, even if they brand themselves with one or the other.

Who Uses Predictive Analytics?

In the 1990s and early 2000s, the use of advanced analytics, referred to as data mining or computational statistics, was relegated to only the most forward-looking companies with deep pockets. Many organizations were still struggling with collecting data, let alone trying to make sense of it through more advanced techniques.

Today, the use of analytics has moved from a niche group in large organizations to being an instrumental component of most mid- to large-sized organizations.

The analytics often begins with business intelligence and moves into predictive analytics as the data matures and the pressure to produce greater benefit from the data increases. Even small organizations, for-profit and non-profit, benefit from predictive analytics now, often using open source software to drive decisions on a small scale.

Challenges in Using Predictive Analytics

Predictive analytics can generate significant improvements in efficiency, decision-making, and return on investment. But predictive analytics isn't always successful and, in all likelihood, the majority of predictive analytics models are never used operationally.

Some of the most common reasons predictive models don't succeed can be grouped into four categories: obstacles in management, obstacles with data, obstacles with modeling, and obstacles in deployment.

Obstacles in Management

To be useful, predictive models have to be deployed. Often, deployment in of itself requires a significant shift in resources for an organization and therefore the project often needs support from management to make the transition from research and development to operational solution. If program management is not a champion of the predictive modeling project and the resulting models, perfectly good models will go unused due to lack of resources and lack of political will to obtain those resources.

For example, suppose an organization is building a fraud detection model to identify transactions that appear to be suspicious and are in need of further investigation. Furthermore, suppose the organization can identify 1,000 transactions per month that should receive further scrutiny from investigators. Processes have to be put into place to distribute the cases to the investigators, and the fraud detection model has to be sufficiently trusted by the investigators for them to follow through and investigate the cases. If management is not fully supportive of the predictive models, these cases may be delivered but end up dead on arrival.

Obstacles with Data

Predictive models require data in the form of a single table or flat file containing rows and columns: two-dimensional data. If the data is stored in transactional databases, keys need to be identified to join the data from the data sources to form the single view or table. Projects can fail before they even begin if the keys don't exist in the tables needed to build the data.

Even if the data can be joined into a single table, if the primary inputs or outputs are not populated sufficiently or consistently, the data is meaningless. For example, consider a customer acquisition model. Predictive models need examples of customers who were contacted and did *not* respond as well as those who were contacted and *did* respond. If active customers are stored in one table and marketing contacts (leads) in a separate table, several problems can thwart modeling efforts. First, unless customer tables include the campaign they were acquired from, it may be impossible to reconstruct the list of leads in a campaign along with the label that the lead responded or didn't respond to the contact.

Second, if customer data, including demographics (age, income, ZIP), is overwritten to keep it up-to-date, and the demographics at the time they were acquired is not retained, a table containing leads as they appeared at the time of the marketing campaign can never be reconstructed. As a simple example, suppose phone numbers are only obtained after the lead converts and becomes a customer. A great predictor of a lead becoming a customer would then be whether the lead has a phone number; this is leakage of future data unknown at the time of the marketing campaign into the modeling data.

Obstacles with Modeling

Perhaps the biggest obstacle to building predictive models from the analyst's perspective is *overfitting*, meaning that the model is too complex, essentially memorizing the training data. The effect of overfitting is twofold: The model performs poorly on new data and the interpretation of the model is unreliable. If care isn't taken in the experimental design of the predictive models, the extent of model overfit isn't known until the model has already been deployed and begins to fail.

A second obstacle with building predictive models occurs when zealous analysts become too ambitious in the kind of model that can be built with the available data and in the timeframe allotted. If they try to "hit a home run" and can't complete the model in the timeframe, no model will be deployed at all. Often a better strategy is to build simpler models first to ensure a model of some value will be ready for deployment. Models can be augmented and improved later if time allows.

For example, consider a customer retention model for a company with an online presence. A zealous modeler may be able to identify thousands of candidate inputs to the retention model, and in an effort to build the best possible model, may be slowed by the sheer combinatorics involved with data preparation and variable selection prior to and during modeling.

However, from the analyst's experience, he or she may be able to identify 100 variables that have been good predictors historically. While the analyst suspects that a better model could be built with more candidate inputs, the first model can be built from the 100 variables in a much shorter timeframe.

Obstacles in Deployment

Predictive modeling projects can fail because of obstacles in the deployment stage of modeling. The models themselves are typically not very complicated computationally, requiring only dozens, hundreds, thousands, or tens of thousands of multiplies and adds, easily handled by today's servers.

At the most fundamental level, however, the models have to be able to be interrogated by the operational system and to issue predictions consistent with that system. In transactional systems, this typically means the model has to be encoded in a programming language that can be called by the system, such as SQL, C++, Java, or another high-level language. If the model cannot be translated or is translated incorrectly, the model is useless operationally.

Sometimes the obstacle is getting the data into the format needed for deployment. If the modeling data required joining several tables to form the single modeling table, deployment must replicate the same joining steps to build the data the models need for scoring. In some transactional systems with disparate data forming the modeling table, complex joins may not be possible in the timeline needed. For example, consider a model that recommends content to be displayed on a web page. If that model needs data from the historic patterns of browsing behavior for a visitor and the page needs to be rendered in less than one second, all of the data pulls and transformations must meet this timeline.

What Educational Background Is Needed to Become a Predictive Modeler?

Conventional wisdom says that predictive modelers need to have an academic background in statistics, mathematics, computer science, or engineering. A degree in one of these fields is best, but without a degree, at a minimum, one should at least have taken statistics or mathematics courses. Historically, one could not get a degree in predictive analytics, data mining, or machine learning.

This has changed, however, and dozens of universities now offer master's degrees in predictive analytics. Additionally, there are many variants of analytics degrees, including master's degrees in data mining, marketing analytics, business analytics, or machine learning. Some programs even include a practicum so that students can learn to apply textbook science to real-world problems.

One reason the real-world experience is so critical for predictive modeling is that the science has tremendous limitations. Most real-world problems have data problems never encountered in the textbooks. The ways in which data can go wrong are seemingly endless; building the same customer acquisition models even within the same domain requires different approaches to data preparation, missing value imputation, feature creation, and even modeling methods.

However, the *principles* of how one can solve data problems are not endless; the experience of building models for several years will prepare modelers to at least be able to identify when potential problems may arise.

Surveys of top-notch predictive modelers reveal a mixed story, however. While many have a science, statistics, or mathematics background, many do not. Many have backgrounds in social science or humanities. How can this be?

Consider a retail example. The retailer Target was building predictive models to identify likely purchase behavior and to incentivize future behavior with relevant offers. Andrew Pole, a Senior Manager of Media and Database Marketing described how the company went about building systems of predictive models at the Predictive Analytics World Conference in 2010. Pole described the importance of a combination of domain knowledge, knowledge of predictive modeling, and most of all, a forensic mindset in successful modeling of what he calls a "guest portrait."

They developed a model to predict if a female customer was pregnant. They noticed patterns of purchase behavior, what he called "nesting" behavior. For example, women were purchasing cribs on average 90 days before the due date. Pole also observed that some products were purchased at regular intervals prior to a woman's due date. The company also observed that if they were able to acquire these women as purchasers of other products during the time before the birth of their baby, Target was able to increase significantly the customer value; these women would continue to purchase from Target after the baby was born based on their purchase behavior before.

The key descriptive terms are *"observed"* and *"noticed."* This means the models were not built as black boxes. The analysts asked, "does this make sense?" and leveraged insights gained from the patterns found in the data to produce better predictive models. It undoubtedly was iterative; as they "noticed" patterns, they were prompted to consider other patterns they had not explicitly considered before (and maybe had not even occurred to them before). This forensic mindset of analysts, noticing interesting patterns and making connections between those patterns and how the models could be used, is critical to successful modeling. It is rare that predictive models can be fully defined before a project and anticipate all of the most important patterns the model will find. So we shouldn't be surprised that we *will* be surprised, or put another way, we should *expect* to be surprised.

This kind of mindset is not learned in a university program; it is part of the personality of the individual. Good predictive modelers need to have a forensic mindset and intellectual curiosity, whether or not they understand the mathematics enough to derive the equations for linear regression.

CHAPTER

2

Setting Up the Problem

The most important part of any predictive modeling project is the very beginning when the predictive modeling project is defined. Setting up a predictive modeling project is a very difficult task because the skills needed to do it well are very broad, requiring knowledge of the business domain, databases, or data infrastructure, and predictive modeling algorithms and techniques. Very few individuals have all of these skill sets, and therefore setting up a predictive modeling project is inevitably a team effort.

This chapter describes principles to use in setting up a predictive modeling project. The role practitioners play in this stage is critical because missteps in defining the unit of analysis, target variables, and metrics to select models can render modeling projects ineffective.

Predictive Analytics Processing Steps: CRISP-DM

The Cross-Industry Standard Process Model for Data Mining (CRISP-DM) describes the data-mining process in six steps. It has been cited as the most-often used process model since its inception in the 1990s. The most frequently cited alternative to CRISP-DM is an organization's or practitioner's own process model, although upon more careful examination, these are also essentially the same as CRISP-DM.

One advantage of using CRISP-DM is that it describes the most commonly applied steps in the process and is documented in an 80-page PDF file. The CRISP-DM name itself calls out data mining as the technology, but the same process model applies to predictive analytics and other related analytics approaches, including business analytics, statistics, and text mining.

The CRISP-DM audience includes both managers and practitioners. For program managers, CRISP-DM describes the steps in the modeling process from a program perspective, revealing the steps analysts will be accomplishing as they build predictive models. Each of the steps can then have its own cost estimates and can be tracked by the manager to ensure the project deliverables and timetables are met. The last step in many of the sub-tasks in CRISP-DM is a report describing what decisions were made and why. In fact, the CRISP-DM document identifies 28 potential deliverables for a project. This certainly is music to the program manager's ears!

For practitioners, the step-by-step process provides structure for analysis and not only reminds the analyst of the steps that need to be accomplished, but also the need for documentation and reporting throughout the process, which is particularly valuable for new modelers. Even for experienced practitioners, CRISP-DM describes the steps succinctly and logically. Many practitioners are hesitant to describe the modeling process in linear, step-by-step terms because projects almost never proceed as planned due to problems with data and modeling; surprises occur in nearly every project. However, a good baseline is still valuable, especially as practitioners describe to managers what they are doing and why they are doing it; CRISP-DM provides the justification for the steps that need to be completed in the process of building models.

The six steps in the CRISP-DM process are shown in Figure 2-1: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. These steps, and the sequence they appear in the figure, represent the most common sequence in a project. These are described briefly in Table 2-1.

Table 2-1: CRISM-DM Sequence

STAGE	DESCRIPTION
Business Understanding	Define the project.
Data Understanding	Examine the data; identify problems in the data.
Data Preparation	Fix problems in the data; create derived variables.
Modeling	Build predictive or descriptive models.
Evaluation	Assess models; report on the expected effects of models.
Deployment	Plan for use of models.

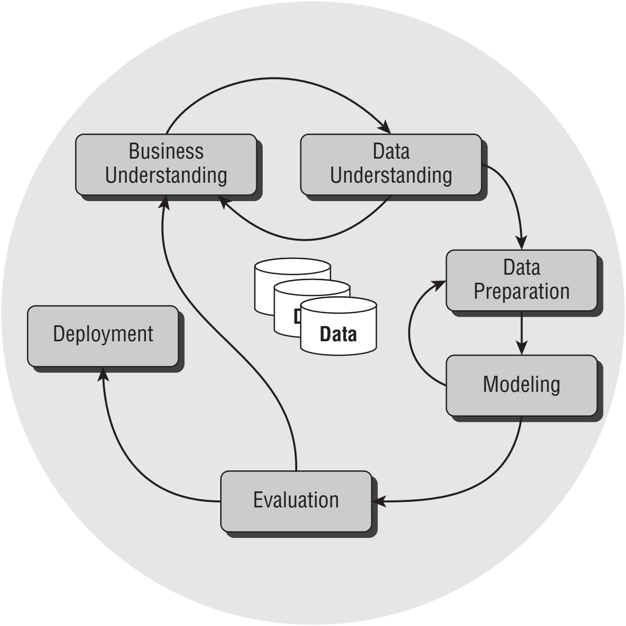

Figure 2-1: The CRISP-DM process model

Note the feedback loops in the figure. These indicate the most common ways the typical process is modified based on findings during the project. For example, if business objectives have been defined during Business Understanding, and then data is examined during Data Understanding, you may find that there is insufficient data quantity or data quality to build predictive models. In this case, Business Objectives must be re-defined with the available data in mind before proceeding to Data Preparation and Modeling. Or consider a model that has been built but has poor accuracy. Revisiting data preparation to create new derived variables is a common step to improve the models.

Business Understanding

Every predictive modeling project needs objectives. Domain experts who understand decisions, alarms, estimates, or reports that provide value to an organization must define these objectives. Analysts themselves sometimes have this expertise, although most often, managers and directors have a far better perspective on how models affect the organization. Without domain expertise, the definitions of what models should be built and how they should be assessed can lead to failed projects that don't address the key business concerns.

The Three-Legged Stool

One way to understand the collaborations that lead to predictive modeling success is to think of a three-legged stool. Each leg is critical to the stool remaining stable and fulfilling its intended purpose. In predictive modeling, the three legs of the stool are (1) domain experts, (2) data or database experts, and (3) predictive modeling experts. Domain experts are needed to frame a problem properly in a way that will provide value to the organization. Data or database experts are needed to identify what data is available for predictive modeling and how that data can be accessed and normalized. Predictive modelers are needed to build the models that achieve the business objectives.

Consider what happens if one or more of these three legs are missing. If the problem is not defined properly and only modelers and the database administrator are defining the problems, excellent models may be built with fantastic accuracy only to go unused because the model doesn't address an actual need of the organization. Or in a more subtle way, perhaps the model predicts the right kind of decision, but the models are assessed in a way that doesn't address very well what matters most to the business; the wrong model is selected because the wrong metric for describing good models is used.

If the database expert is not involved, data problems may ensue. First, there may not be enough understanding of the layout of tables in the database to be able to access all of the fields necessary for predictive modeling. Second, there may be insufficient understanding of fields and what information they represent even if the names of the fields seem intuitive, or worse still, if the names are cryptic and no data dictionary is available. Third, insufficient permissions may preclude pulling data into the predictive modeling environment. Fourth, database resources may not support the kinds of joins the analyst may believe he or she needs to build the modeling data. And fifth, model deployment options envisioned by the predictive modeling team may not be supported by the organization.

If the predictive modelers are not available during the business understanding stage of CRISP-DM, obstacles outlined in this chapter may result. First, a lack of understanding by program managers of what the predictive models can do, driven by hype around predictive modeling, can lead the manager to specify models that are impossible to actually build. Second, defining target variables for predictive modeling may not be undertaken at all or, if done, may be specified poorly, thwarting predictive modeling efforts. Third, without predictive modelers defining the layout of data needed for building predictive models, a modeling table to be used by the modeler may not be defined at all or may lack key fields needed for the models.

Business Objectives

Assuming all three types of individuals that make up the three-legged stool of predictive modeling are present during the Business Understand stage of CRISP-DM, tradeoffs and compromises are not unusual during the hours or even days of meetings that these individuals and groups participate in so that solid business and predictive modeling objectives are defined.

Six key issues that should be resolved during the Business Understanding stage include definitions of the following:

- Core business objectives to be addressed by the predictive models
- How the business objectives can be quantified
- What data is available to quantify the business objectives
- What modeling methods can be invoked to describe or predict the business objectives
- How the goodness of model fit of the business objectives are quantified so that the model scores make business sense
- How the predictive models can be deployed operationally

Frequently, the compromises reached during discussions are the result of the imperfect environment that is typical in most organizations. For example, data that you would want to use in the predictive models may not be available in a timely manner or at all. Target variables that address the business objectives more directly may not exist or be able to be quantified. Computing resources may not exist to build predictive models in the way the analysts would prefer. Or there may not be available staff to apply to the project in the timeframe needed. And these are just a few possible issues that may be uncovered. Project managers need to be realistic about which business objectives can be achieved in the timeframe and within the budget available.

Predictive modeling covers a wide range of business objectives. Even the term "business objectives" is restrictive as modeling can be done for more than just what you normally associate with a commercial enterprise. Following is a short list of predictive modeling projects. I personally have either built models for, or advised a customer on, building models for each of these projects.

PROJECT		
Customer acquisition/ Response/Lead generation	Credit card application fraud	Medical image anomaly detection
Cross-sell/Up-sell	Loan application fraud	Radar signal, vehicle/aircraft identification
Customer next product to purchase	Invoice fraud	Radar, friend-or-foe differentiation
Customer likelihood to purchase in *N* days	Insurance claim fraud	Sonar signal object identification (long and short range)
Website—next site to interact with	Insurance application fraud	Optimum guidance commands for smart bombs or tank shells
Market-basket analysis	Medical billing fraud	Likelihood for flight to be on time
Customer value/Customer profitability	Payment fraud	Insurance risk of catastrophic claim
Customer segmentation	Warranty fraud	Weed tolerance to pesticides
Customer engagement with brand	Tax collection likelihood to pay	Mean time to failure/ Likelihood to fail
Customer attrition/Retention	Non-filer predicted tax liability	Likelihood of hardware failure due to complexity
Customer days to next purchase	Patient likelihood to re-admit	Fault detection/Fault explanation
Customer satisfaction	Patient likelihood to comply with medication protocols	Part needed for repair
Customer sentiment/ Recommend to a friend	Cancer detection	Intrusion detection/ Likelihood of an intrusion event
Best marketing creative	Gene expression/ Identification	New hire likelihood to succeed/advance
Credit card transaction fraud	Predicted toxicity (LD50 or LC50) of substance	New hire most desirable characteristics

While many models are built to predict the behavior of people or things, not all are. Some models are built expressly for the purpose of understanding the behavior of people, things, or processes better. For example, predicting "weed tolerance to pesticides" was built to test the hypothesis that the weeds were becoming intolerant to a specific pesticide. The model identified the primary contributors in predicting success or failure in killing the weeds; this in and of itself was insightful. While the likelihood of a customer purchasing a product within seven days is interesting on its own, understanding *why* the customer is likely to purchase can provide even more value as the business decides how best to contact the individuals. Or if a Customer Retention model is built with high accuracy, those customers that match the profile for retention but attrite nevertheless become a good segment for call-center follow-up.

Defining Data for Predictive Modeling

Data for predictive modeling must be two-dimensional, comprised of rows and columns. Each row represents what can be called a *unit of analysis*. For customer analytics, this is typically a customer. For fraud detection, this may be a transaction. For call center analytics, this may refer to an individual call. For survey analysis, this may be a single survey. The unit of analysis is problem-specific and therefore is defined as part of the Business Understanding stage of predictive modeling.

If data for modeling is loaded from files, the actual form of the data is largely irrelevant because most software packages support data in a variety of formats:

- Delimited flat files, usually delimited with commas (`.csv` files), tabs, or some other custom character to indicate where field values begin and end
- Fixed-width flat files with a fixed number of characters per field. No delimiters are needed in this format but the exact format for each field must be known before loading the data.
- Other customized flat files
- Binary files, including formats specific to software packages such as SPSS files `(.sav`), SAS files (`.sas7bdat`), and Matlab files (`.mat`)

Most software packages also provide connectivity to databases through native or ODBC drivers so that tables and views can be accessed directly from the software. Some software allows for the writing of simple or even complex queries to access the data from within the software itself, which is very convenient for several reasons:

- Data does not have to be saved to disk and loaded into the predictive modeling software, a slow process for large data sets.
- Data can be maintained in the database without having to provide version control for the flat files.
- Analysts have greater control and flexibility over the data they pull from the database or data mart.

However, you must also be careful that the tables and views being accessed remain the same throughout the modeling project and aren't changing without any warning. When data changes without the knowledge of the analyst, models can also change inexplicably.

Defining the Columns as Measures

Columns in the data are often called *attributes*, *descriptors*, *variables*, *fields*, *features*, or just columns. The book will use these labels interchangeably. Variables in the data are measures that relate to or describe the record. For customer analytics, one attribute may be the customer ID, a second the customer's age, a third the customer's street address, and so forth. The number of attributes in the data is limited only by what is measured for the particular unit of analysis, which attributes are considered to be useful, and how many attributes can be handled by the database or predictive modeling software.

Columns in the data are measures of that unit of analysis, and for predictive modeling algorithms, the number of columns and the order of the columns must be identical from record to record. Another way to describe this kind of data is that the data is rectangular. Moreover, the meaning of the columns must be consistent. If you are building models based on customer behavior, you are faced with an immediate dilemma: How can you handle customers who have visited different numbers of times and maintain the rectangular shape of the data?

Consider Table 2-2 with two customers of a hotel chain, one of whom has visited three times and the other only once. In the table layout, the column labeled "Date of Visit 1" is the date of the first visit the customer made to the hotel property. Customer 100001 has visited only once and therefore has no values for visit 2 and visit 3. These can be labeled as "NULL" or just left blank. The fact that they are not defined, however, can cause problems for some modeling algorithms, and therefore you often will not represent multiple visits as separate columns. This issue is addressed in Chapter 4.

Table 2-2: Simple Rectangular Layout of Data

CUSTOMER ID	DATE OF VISIT 1	DATE OF VISIT 2	DATE OF VISIT 3
100001	5/2/12	NULL	NULL
100002	6/9/12	9/29/12	10/13/12

There is a second potential problem with this layout of the data, however. The "Date of Visit 1" is the first visit. What if the pattern of behavior related to the models is better represented by how the customer behaved most recently? For customer 100001, the most recent visit is contained in the column "Date of

Visit 1," whereas for customer 100002, the most recent visit is in the column "Date of Visit 3." Predictive modeling algorithms consider each column as a separate measure, and therefore, if there is a strong pattern related to the most recent visit, the pattern is broken in this representation of the data. Alternatively, you could represent the same data as shown in Table 2-3.

Table 2-3: Alternative Rectangular Layout of Data

CUSTOMER ID	DATE OF VISIT 1	DATE OF VISIT 2	DATE OF VISIT 3
100001	5/2/12	NULL	NULL
100002	10/13/12	9/29/12	6/9/12

In this data, Visit 1 is no longer the first visit but is the most recent visit, Visit 2 is two visits ago, Visit 3 is three visits ago, and so on. The representation of the data you choose is dependent on which representation is expected to provide the most predictive set of inputs to models.

A third option for this customer data is to remove the temporal data completely and represent the visits in a consistent set of attributes that summarizes the visits. Table 2-4 shows one such representation: The same two customers are described by their most recent visit, the first visit, and the number of visits.

Table 2-4: Summarized Representation of Visits

CUSTOMER ID	DATE OF FIRST VISIT	DATE OF MOST RECENT VISIT	NUMBER OF VISITS
100001	5/2/12	5/2/12	1
100002	6/9/12	10/13/12	3

Ultimately, the representation problems described in Tables 2-3, 2-4, and 2-5 occur because this data is inherently three dimensional, not two. There is a temporal dimension that has to be represented in the row-column format, usually by summarizing the temporal dimension into *features* of the data.

Defining the Unit of Analysis

Predictive modeling algorithms assume that each record is an independent observation. Independence in this context merely means that the algorithms do not consider direct connections between records. For example, if records 1 and 2 are customers who are husband and wife and frequently travel together, the algorithms treat these two no differently than any two people with similar patterns of behavior; the algorithms don't know they are related or connected.

If the rows are not independent and in some way are connected to each other, the data itself will not be representative of the broad set of patterns in the complete set of records the model may encounter after it is deployed.

Consider hospitality analytics where each record is a customer who visits the property of a hotel. The assumption of independence is satisfied because (presumably) each customer behaves as a separate, independent entity. This is plausible, although there are exceptions to this assumption in the case of business conventions and conferences.

But what if the modeling data is not truly independent and is built from a single organization that has a contract with the hotel. Assume also that the organization has travel procedures in place so that reservations are always placed in the same way, and visits can only be made for business purposes. These records have a connection with each other, namely in the procedures of the organization. A model built from these records is, of course, biased because the data comes from a single organization, and therefore may not apply well to the general population. In addition, however, models built from this data will identify patterns that are related to the organization's procedures rather than the visitors' behavior had they decided on their own to visit the hotel.

Which Unit of Analysis?

It isn't always clear which unit of analysis to use. Suppose you are building models for the same hospitality organization and that the business objectives include identifying customer behavior so they can customize marketing creative content to better match the type of visitor they are contacting. Assume also that the objective is to increase the number of visits a customer makes to the property in the next quarter.

The first obvious choice is to make the unit of analysis a person: a customer. In this scenario, each record will represent an individual customer, and columns will describe the behavior of that customer, including their demographics and their behavior at the hotel. If the customer has visited the hotel on multiple occasions, that history must be represented in the single record. Some possible variables relate to the most recent visit, how frequently the customer visits, and how much the customer spends on the reservation, restaurant, and other amenities at the hotel. These derived variables will take some effort to define and compute. However, each record will contain a complete summary of that customer's behavior, and no other record will relate to that particular customer—each record is an independent observation.

What is obscured by this representation is detail-related to each individual visit. If a customer has undergone a sequence of visits that is trending toward higher or lower spend, rolling up the visits into a single record for the customer will obscure the details unless the details are explicitly revealed as columns in the data. To capture specific detail, you must create additional derived variables.

A second unit of analysis is the visit: Each record contains only information about a single visit and the behavior of the customer during that visit. If a customer visited ten times, that customer will have ten records in the data and these records would be considered independent events without any immediate connection to one another; the modeling algorithms would not know that this particular customer had visited multiple times, nor would the algorithms know there is a connection between the visits. The effect of an individual customer visiting multiple times is this: The pattern of that customer is weighted by the number of times the customer visits, making it appear to the algorithms that the pattern exists more often in the data. From a visit perspective, of course, this is true, whereas from a customer perspective it is not true.

One way to communicate the connection between visits to individual customers is to include derived variables in the data that explicitly connect the history of visits. Derived variables such as the number of visits, the average spend in the past three visits, and the number of days since the last visit can all be added. You must take care to avoid leaking future information from the derived variable into the record for the visit, however. If you create a derived variable summarizing the number of visits the customer has made, no future visit can be included, only visits whose end dates precede the date of the visit for the record. This precludes creating these derived variables from simple "group by" operations with an appropriate "where" clause.

Ultimately, the unit of analysis selected for modeling is determined by the business objectives and how the model will be used operationally. Are decisions made from the model scores based on a transaction? Are they made based on the behavior of a single customer? Are they made based on a single visit, or based on aggregate behavior of several transactions or visits over a time period? Some organizations even build multiple models from the same data with different units of analysis precisely because the unit of analysis drives the decisions you can make from the model.

Defining the Target Variable

For models that estimate or predict a specific value, a necessary step in the Business Understanding stage is to identify one or more *target variables* to predict. A target variable is a column in the modeling data that contains the values to be estimated or predicted as defined in the business objectives. The target variable can be numeric or categorical depending on the type of model that will be built.

Table 2-5 shows possible target variables associated with a few projects from the list of projects in the section "Business Objectives" that appears earlier in the chapter.

Table 2-5: Potential Target Variables

PROJECT	TARGET VARIABLE TYPE	TARGET VARIABLE VALUES
Customer Acquisition	Binary	1 for acquired, 0 for non-acquired
Customer Value	Continuous	Dollar value of customer
Invoice Fraud Detection	Categorical	Type of fraud (5 levels)
Days to Next Purchase	Continuous	Number of days
Days to Next Purchase <= 7	Binary	1 for purchased within 7 days, 0 for did not purchase within 7 days

The first two items in the table are typical predictive modeling problems: the first a classification problem and the second an estimation problem. These will be defined in Chapter 8. The third item, Invoice Fraud Detection, could have been defined with a binary target variable (1 for "fraud," 0 for "not fraud") but was instead defined with five levels: four types of fraud and one level for "not fraud." This provides not only an indication as to whether or not the invoice is fraudulent, but also a more specific prediction of the type of fraud that could be used by the organization in determining the best course of action.

Note the last two items in the table. Both target variables address the same idea, predicting the number of days until the next purchase. However, they are addressed in different ways. The first is the more straightforward prediction of the actual number of days until the next purchase although some organizations may want to constrain the prediction to a 7-day window for a variety of reasons. First, they may not care if a customer will purchase 30 or 60 days from now because it is outside of the window of influence they may have in their programs. Second, binary classification is generally an easier problem to solve accurately. These models do not have to differentiate between someone who purchases 14 days from now from someone who purchases 20 days from now: In the binary target variable formulation, these are the same (both have value 0). The predictive models may predict the binary target variable more accurately than the entire distribution.

Defining the target variable is critically important in a predictive modeling project because it is the only information the modeling algorithms have to uncover what the modeler and program manager desire from the predictions. Algorithms do not have common sense and do not bring context to the problem in the way the modeler and program manager can. The target variable definition therefore must describe or quantify as much as possible the business objective itself.

Temporal Considerations for Target Variable

For most modeling projects focused on predicting future behavior, careful consideration for the timeline is essential. Predictive modeling data, as is the case with all data, is historical, meaning that it was collected in the past. To build a model that predicts so-called future actions from historic data requires shifting the timeline in the data itself.

Figure 2-2 shows a conceptualized timeline for defining the target variable. If the date the data is pulled is the last vertical line on the right, the "Data pull timestamp," all data used for modeling by definition must precede that date. Because the timestamp for the definition of the target variable value must occur *after* the last information known from the input variables, the timeline for constructing the modeling data must be shifted to the left.

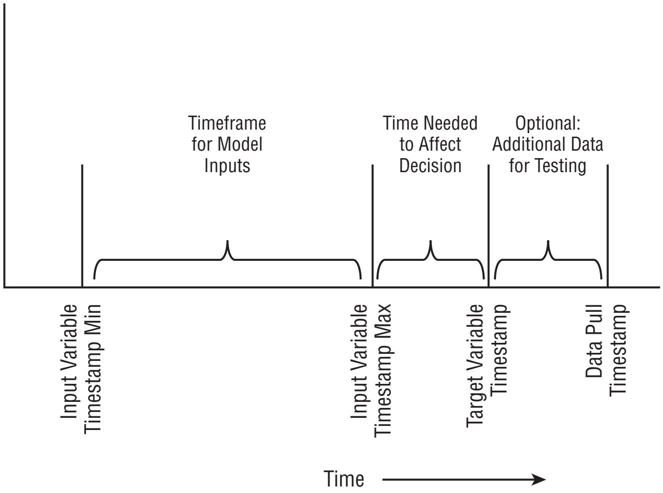
Timeframe for Model Inputs
Time Needed to Affect Decision
Optional: Additional Data for Testing
Input Variable Timestamp Min
Input Variable Timestamp Max
Target Variable Timestamp
Data Pull Timestamp
Time

Figure 2-2: Timeline for defining target variable

The "Time Needed to Affect Decision" time gap can be critical in models, allowing time for a treatment to mature. For example, if you are building models to identify churn, the lead time to predict when churn might take place is critical to putting churn mitigation programs in place. You might, therefore, want to predict if churn will occur 30–60 days in the future, in which case there must be a 30-day gap between the most recent input variable timestamp and the churn timestamp.

The "Timeframe for Model Inputs" range has two endpoints. The "Input Variable Timestamp Max" is defined by the target variable and time needed to affect the decision. However, the minimum time is defined by the business objectives and practical considerations. Sometimes maintaining a long temporal sequence of events is difficult and costly. Other times, the perception of domain experts is that information more than several years old is too stale and unlikely to be valuable in predicting the target variable.

The time gap at the right, "Additional Data for Testing," can sometimes be valuable during model validation. The best validation data are data that have not been seen by the modeling algorithms and emulate what the model will be doing when it goes live. Data collected subsequent to the target variable timestamp is certainly held-out data. If the model is biased by the particular timeframe when the target variable was computed or created, predictions for data collected after the target variable definition will reveal the bias.

Defining Measures of Success for Predictive Models

The determination of what is considered a good model depends on the particular interests of the organization and is specified as the business success criterion. The business success criterion needs to be converted to a predictive modeling criterion so the modeler can use it for selecting models.

If the purpose of the model is to provide highly accurate predictions or decisions to be used by the business, measures of accuracy will be used. If interpretation of the business is what is of most interest, accuracy measures will not be used; instead, subjective measures of what provides maximum insight may be most desirable. Some projects may use a combination of both so that the most accurate model is not selected if a less accurate but more transparent model with nearly the same accuracy is available.

Success criteria for classification and estimation models is described in more detail in Chapter 9.

Success Criteria for Classification

For classification problems, the most frequent metrics to assess model accuracy is Percent Correct Classification (PCC). PCC measures overall accuracy without regard to what kind of errors are made; every error has the same weight. Another measure of classification accuracy is the confusion matrix, which enumerates different ways errors are made, like false alarms and false dismissals. PCC and the confusion matrix metrics are good when an entire population must be scored and acted on. For example, if customers who visit a website are to be served customized content on the site based on their browsing behavior, every visitor will need a model score and a treatment based on that score.

If you are treating a subset of the population, a *selected population*, sorting the population by model score and acting on only a portion of those entities in the selected group can be accomplished through metrics such as Lift, Gain, ROC, and Area under the Curve (AUC). These are popular in customer analytics where the model selects a subpopulation to contact with a marketing message, or in fraud analytics, when the model identifies transactions that are good candidates for further investigation.

Success Criteria for Estimation

For continuous-valued estimation problems, metrics often used for assessing models are R^2, average error, Mean Squared Error (MSE), median error, average absolute error, and median absolute error. In each of these metrics, you first compute the error of an estimate—the actual value minus the predicted estimate—and then compute the appropriate statistic based on those errors. The values are then summed over all the records in the data.

Average errors can be useful in determining whether the models are biased toward positive or negative errors. Average absolute errors are useful in estimating the magnitude of the errors (whether positive or negative). Analysts most often examine not only the overall value of the success criterion, but also examine the entire range of predicted values by creating scatterplots of actual target values versus predicted values or actual target values versus residuals (errors).

In principle, you can also include rank-ordered metrics such as AUC and Gain as candidates to estimate the success criteria, although they often are not included in predictive analytics software for estimation problems. In these instances, you need to create a customized success criterion.

Other Customized Success Criteria

Sometimes none of the typical success criteria are sufficient to evaluate predictive models because they do not match the business objective. Consider the invoice fraud example described earlier. Let's assume that the purpose of the model is to identify 100 invoices per month to investigate from the hundreds of thousands of invoices submitted. If the analyst builds a classification model and selects the model that maximizes PCC, the analyst can be fooled into thinking that the best model as assessed by PCC is good, even though none of the top 100 invoices are good candidates for investigation. How is this possible? If there are 100,000 invoices submitted in a month, you are selecting only 0.1 percent of them for investigation. The model could be perfect for 99.9 percent of the population and miss what you care about the most, the top 100.

In situations where there are specific needs of the organization that lead to building models, it may be best to consider customized cost functions. In the fraud example, you want to identify a population of 100 invoices that is maximally productive for the investigators. If the worst scenario for the investigators is to pursue a false alarm, a case that turns out to not be fraudulent at all, the model should reflect this cost in the ranking. What modeling metric does this? No metric addresses this directly, although ROC curves are close to the idea. You could therefore select models that maximize the area under the ROC curve at the depth of 100 invoices. However, this method considers true alerts and false alarms as equally positive or negative. One solution is to consider the cost of false alarms greater than the benefit of a true alert; you can penalize false alarms ten times as much as a true alert. The actual cost values are domain-specific, derived either empirically or defined by domain experts.

Another candidate for customized scoring of models includes Return On Investment (ROI) or profit, where there is a fixed or variable cost associated with the treatment of a customer or transaction (a record in the data), and a fixed or variable return or benefit if the customer responds favorably. For example, if you are building a customer acquisition model, the cost is typically a fixed cost associated with mailing or calling the individual; the return is the estimated value of acquiring a new customer. For fraud detection, there is a cost associated with investigating the invoice or claim, and a gain associated with the successful recovery of the fraudulent dollar amount.

Note that for many customized success criteria, the actual predicted values are not nearly as important as the rank order of the predicted values. If you compute the cumulative net revenue as a customized cost function associated with a model, the predicted probability may never enter into the final report, except as a means to threshold the population into the "select" group (that is to be treated) and the "nonselect" group.

Doing Predictive Modeling Out of Order

While CRISP-DM is a useful guide to follow for predictive modeling projects, sometimes there are advantages to deviating from the step-by-step structure to obtain insights more quickly in a modeling project. Two useful deviations are to build models first and to deploy models before they are completed.

Building Models First

One of the biggest advantages of predictive modeling compared to other ways to analyze data is the automation that occurs naturally in many algorithms. The automation allows you to include many more input variables as candidates for inclusion in models than you could with handcrafted models. This is particularly the case with decision trees because, as algorithms, they require minimal data preparation compared to other algorithms.

Building models before Data Preparation has been completed, and sometimes even before Data Understanding has been undertaken, is usually problematic. However, there are also advantages to building models. You can think of it as an additional step in Data Understanding.

First, the predictive models will give the modeler a sense for which variables are good predictors. Some variables that would be good predictors if they were prepared properly would not, of course, show up. Second, building predictive models early gives the modeler a sense of what accuracy can be expected without applying any additional effort: a baseline.

Third, and perhaps one of the biggest insights that can be gained from building models early, is to identify models that predict too well. If models predict perfectly or much better than expected, it is an indication that an input variable contains future information subsequent to the target variable definition or contains information about the target variable itself. Sometimes this is obvious; if the address of a customer is only known after they respond to a mailing and the customer's state is included as an input, state values equal to NULL will always be related to non-responders. But sometimes the effect is far more subtle and requires investigation by the modeler to determine why a variable is inexplicably a great predictor.

Early Model Deployment

Model deployment can take considerable effort for an organization, involving individuals from multiple departments. If the model is to be deployed as part of a real-time transactional system, it will need to be integrated with real-time data feeds.

The Modeling stage of CRISP-DM is iterative and may take weeks to months to complete. However, even early in the process, the modeler often knows which variables will be the largest contributors to the final models and how those variables need to be prepared for use in the models. Giving these specifications and early versions of the models to the deployment team can aid in identifying potential obstacles to deployment once the final models are built.

Case Study: Recovering Lapsed Donors

This case study provides an example of how the business understanding steps outlined in this chapter can be applied to a particular problem. This particular case study, and data from the case study, will be used throughout the book for examples and illustrations.

Overview

The KDD-Cup is a data competition that became part of the annual Knowledge Discovery and Data Mining (KDD) conference. In the competition, one or more data sets, including record ID, inputs, and target variables, are made available to any who wish to participate. The 1998 KDD-Cup competition was a non-profit donation-modeling problem.

Business Objectives

The business objective was to recover lapsed donors. When donors gave at least one gift in the past year (0–12 months ago), they were considered active donors. If they did not give a gift in the past year but did give 13–24 months ago, they were considered lapsed donors. Of the lapsed donors, could you identify characteristics based on their historical patterns of behavior with the non-profit organization? If these donors could be identified, then these lapsed donors could be solicited again and lapsed donors unlikely to give again could be ignored, increasing the revenue to the organization.

The test mailing that formed the basis for the competition had already been completed so considerable information about the recovery of lapsed donors was already known. The average donation amount for all lapsed donors who were mailed was $0.79 and the cost of the mail campaign was $0.68 per contact. So soliciting everyone was still profitable, and the amount of profit was approximately $11,000 per 100,000 lapsed donors contacted.

But could this be improved? If a predictive model could provide a score, where a higher score means the donor is more likely to be a good donor, you could rank the donors and identify the best, most profitable subset.

Data for the Competition

The recovery concept was to mail to a random subset of lapsed donors (191,779 of them). Approximately 5.1 percent of them responded to the mailing. The unit of analysis was a donor, so each record was a lapsed donor, and the attributes associated with each donor included demographic information and historic patterns of giving with the non-profit organization. Many derived variables were included in the data, including measures that summarized their giving behavior (recent, minimum, maximum, and average gift amounts), RFM snapshots at the time of each of the prior contacts with each donor, their socioeconomic status, and more.

The Target Variables

Two target variables were identified for use in modeling: TARGET_B and TARGET_D. Responders to the recovery mailing were assigned a value of 1 for

TARGET_B. If the lapsed donor did not respond to the mailing, he received a TARGET_B value of 0. TARGET_D was populated with the amount of the gift that the lapsed donor gave as a result of the mailing. If he did not give a gift at all (i.e., TARGET_B = 0), the lapsed donor was assigned a value of 0 for TARGET_D.

Thus, there are at least two kinds of models that can be built. If TARGET_B is the target variable, the model will be a binary classification model to predict the likelihood a lapsed donor can be recovered with a single mailing. If TARGET_D is the target variable, the model predicts the amount a lapsed donor gives from a single recovery campaign.

Modeling Objectives

The next step in the Business Understanding process is translating the business objective—maximizing net revenue—to a predictive modeling objective. This is a critical step in the process because, very commonly, the business objective does not translate easily or directly to a quantitative measure.

What kinds of models can be built? There are two identifying measures that are candidate target variables. First, TARGET_B is the response indicator: 1 if the donor responded to the mailing and 0 if the donor did not respond to the mailing. Second, TARGET_D is the amount of the donation if the donor responded. The value of TARGET_D is $0 if the donor did not respond to the mailing.

If a TARGET_B model is built, the output of the prediction (the Score) is the probability that a donor will respond. The problem with this number, however, is that it doesn't address directly the amount of a donation, so it won't address directly net revenue. In fact, it considers all donors equally (they all are coded with a value of 1 regardless of how much they gave) and it is well known that donation amounts are inversely correlated with likelihood to respond. These models therefore would likely favor low-dollar, lower net revenue donors.

On the other hand, a TARGET_D model appears to address the problem head on: It predicts the amount of the donation and therefore would rank the donors by their predicted donation amount and, by extension, their net revenue (all donors have the same fixed cost). But TARGET_D models have a different problem: You only know the amount given for those donors who gave. The vast majority of donors did not respond and therefore have a value of TARGET_D equal to $0. This is a very difficult distribution to model: It has a spike at $0 and a much smaller, heavily skewed distribution for values greater than $0. Typically in these kinds of problems, you would create a model for only those who donated, meaning those with TARGET_B = 1 or, equivalently, those with TARGET_D greater than 0.

This introduces a second problem: If you build a model for just the subset of those who donated, and the model would be applied to the entire population of lapsed donors, are you sure that the predicted values would apply well to those who don't donate? The model built this way is only built to predict the donation amount of donors who actually gave.

This problem therefore is the classic *censored regression* problem where both the selection stage (TARGET_B) and the amount stage (TARGET_D) need to be modeled so that a complete prediction of an expected donation amount can be created.

One solution, though certainly not the only one, would be to build both TARGET_B and TARGET_D models, and then multiply their prediction values to estimate the expected donation amount. The TARGET_B prediction is the likelihood that the donor will give, and the TARGET_D prediction is the amount the donor will give. After multiplying, if a donor is very likely to give and is likely to give in a large amount, that donor will be at the top of the list. Donors will be considered equally likely to give if, for example, their propensity to give is 0.1 (10 percent likelihood) and the predicted donation amount is $10 or their propensity is 0.05 (5 percent likelihood) and their predicted donation amount is $20. In both cases, the score is 1.0.

Score = P(TARGET_B = 1) × Estimated TARGET_D

Model Selection and Evaluation Criteria

After building models to predict TARGET_B or TARGET_D, how do you determine which model is best? The answer is to use a metric that matches as closely as possible to the business objective. Recall that the business objective is to maximize cumulative net revenue. If you build a model to predict TARGET_B, computing percent correct classification is a measure of accuracy, but won't necessarily select the model that best matches the business objective. Computing gain or lift is closer, but still doesn't necessarily match the business objective. But if you instead compute the cumulative net revenue, the business objective itself, and select the model that maximizes the cumulative net revenue, you will have found the best model according to the business objectives.

The procedure would be this:

1. Score all records by the multiplicative method, multiplying the TARGET_B prediction by the TARGET_D prediction.
2. Rank the scores from largest to smallest.
3. Compute the net revenue for each donor, which is the actual TARGET_D value minus $0.68.
4. Sum the net revenue value as you go down the ranked list.
5. When the net revenue is maximized, this is by definition the maximum cumulative net revenue.
6. The model score associated with the record that generated the maximum cumulative net revenue is the score you should mail to in subsequent campaigns.

Note that once you compute a score for each record based on the model predictions, the scores themselves are not used at all in the computation of cumulative net revenue.

Model Deployment

In subsequent mailings to identify lapsed donors to try to recover, you only need to generate model scores and compare the scores to the threshold defined in the model evaluation criterion defined earlier.

Case Study: Fraud Detection

This case study provides an example of how the business understanding steps outlined in this chapter can be applied to invoice fraud detection. It was an actual project, but the name of the organization has been removed upon their request. Some details have been changed to further mask the organization.

Overview

An organization assessed invoices for payment of services. Some of the invoices were submitted fraudulently. The organization could afford to investigate a small number of invoices in detail. Predictive modeling was proposed to identify key invoices to investigate.

Business Objectives

In this invoice fraud example, the very definition of fraud is key to the modeling process. Two definitions are often considered in fraud detection. The first definition is the strict one, labeling an invoice as fraudulent if and only if the case has been prosecuted and the payee of the invoice has been convicted of fraud. The second definition is looser, labeling an invoice as fraudulent if the invoice has been identified as being worthy of further investigation by one or more managers or agents. In the second definition, the invoice has failed the "smell test," but there is no proof yet that the invoice is fraudulent.

Note that there are advantages and disadvantages to each option. The primary advantage of the first definition is clarity; all of those invoices labeled as fraud have been proven to be fraudulent. However, there are also several disadvantages. First, some invoices may have been fraudulent, but they did not meet the standard for a successful prosecution. Some may have been dismissed on technicalities. Others may have had potential but were too complex to prosecute efficiently and therefore were dropped. Still others may have shown potential, but the agency did not have sufficient resources to complete the investigation.

On the other hand, if you use the second definition, many cases labeled as fraud may not be fraudulent after all, even if they appear suspicious upon first glance. In other words, some “fraudulent” labels are created prematurely; if we had waited long enough for the case to proceed, it would have been clear that the invoice was not fraudulent after all.

In this project, the final determination was made that the strict definition should be used.

Data for the Project

Hundreds of thousands of invoices were available for use in modeling, more than enough for building predictive models. However, there were relatively few labeled fraud cases because of the strict definition of fraud. Moreover, just because an invoice was not labeled as being fraudulent didn’t mean that the invoice was definitively not fraudulent. Undoubtedly, there were invoices labeled as “not fraud” that were actually fraudulent. The problem is that the modeling algorithms don’t know this and believe the labels of fraud and non-fraud are completely accurate.

Let’s assume that the invoice fraud rate was 1 percent. This means that 1 of every 100 invoices is fraudulent. Let’s assume, too, that only half of the fraudulent invoices are identified and prosecuted, leaving the other half unprosecuted. For every 100,000 invoices, 500 are identified as fraudulent and 500 are left in the data with the label “0” even though they are fraudulent. These mislabeled cases can at best confuse models and at worst cause models to miss patterns associated with fraud cases because so many invoices with the same pattern of behavior are also associated with non-fraud invoices.

Modelers, during the Business Understanding stage of the project, determined that the non-fraud population should be sampled so that the likelihood that a mislabeled invoice would be included in the data was greatly reduced.

The Target Variables

The most obvious target variable is the indicator that an invoice is fraudulent (1) or not fraudulent (0). The organization decided, however, to be more precise in their definition by identifying six different kinds of fraud, each of which would be investigated differently. After an initial cluster analysis of the fraud types, it was determined that two of the six fraud types actually overlapped considerably with other fraud types, and therefore the number of fraud types used in modeling was reduced to four. The final target variable therefore was a five-level categorical variable.

Modeling Objectives

The models for this project were to be multi-valued classification models with four fraud classes and one non-fraud class. Misclassification costs were set up when possible—not all algorithms support misclassification costs—so that misclassifying non-fraud invoices as fraud received four times the weight as the converse. Moreover, misclassifying one fraud type as another fraud type did not generate any additional penalty.

Model Selection and Evaluation Criteria

Models were selected to identify the 100 riskiest invoices each month. One key concern was that the workload generated by the model had to be productive for the investigators, meaning that there were few false alarms that the investigators would ultimately be wasting their time pursuing. Therefore, a customized cost function was used to trade off true alerts with false alarms. False alarms were given four times the weight of true alerts. The model that was chosen was the one that performed best on the top 100 invoices it would have selected. Scores were computed according to the following process:

1. Compute a score for each invoice where the score is the maximum of the four probabilities.
2. Sort the scores from high to low.
3. Select the top 100 scores.
4. If the invoice in each record was fraudulent, give the record a score of +1. If the record was not fraudulent, give the record a score of –4.
5. Add the +1 or –4 values for each record. The highest score wins.

With a 4:1 ratio in scoring false alarms to true alerts, if 80 of the top 100 invoices scored by the maximum probability were identified as fraudulent, the weighted score is 0. The actual score, however, does not indicate how well the models predicted fraud, although you could look at the individual probabilities to report which fraud type the invoice was flagged for and how large the probability for the record was.

Model Deployment

Deployment of the model was simple: Each month, the model generated scores for each invoice. The 100 top-scoring invoices were flagged and sent to investigators to determine if they were potentially fraudulent, and if so, they were prosecuted judicially.

Summary

Setting up predictive modeling problems requires knowledge of the business objectives, how to build the data, and how to match modeling objectives to the business objectives so the best model is built. Modelers need to be a part of setting up the problem to ensure that the data and model evaluation criteria are realistic. Finally, don't be afraid to revisit and modify business objectives or modeling objectives as the project unfolds; lessons learned during the modeling process can sometimes reveal unexpected problems with data or new opportunities for improving decisions.