Chapter 8

Descriptive and Predictive Analytics

Modelling is what we most often think of when we think of data mining. Modelling is the process of taking some data (usually) and building a simplified description of the processes that might have generated it. The description is often a computer program or mathematical formula. A model captures the knowledge exhibited by the data and encodes it in some language. Often the aim is to address a specific problem through modelling the world in some form and then use the model to develop a better understanding of the world.

We now turn our attention to building models. As in any data mining project, building models is usually the aim, yet we spend a lot more time understanding the business problem and the data, and working the data into shape, before we can begin building the models. Often we gain much valuable knowledge from our preparation for modelling, and some data mining projects finish at that stage, even without the need to build a model—that might be unusual, though, and we do need to expect to build a model or two. As we will find, we build models early on in a project, then work on our data some more to transform, shape, and clean it, build more models, then return to processing the data once again, and so on for many iterations. Each cycle takes us a step closer to achieving our desired outcomes.

This chapter introduces the concept of models and model builders that fall into the categories of data mining: descriptive and predictive. In this chapter, we provide an overview of these approaches. For descrip-

G. Williams, *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*, Use R, DOI 10.1007/978-1-4419-9890-3_8, © Springer Science+Business Media, LLC 2011

tive data mining, we present cluster analysis and association rules as two approaches to model building. For predictive data mining, we consider both classification and regression models, introducing algorithms like decision trees, random forests, boosting, support vector machines, linear regression, and neural networks. In each case, in their own chapters, the algorithms are presented together with a guide to using them within Rattle and R.

8.1 Model Nomenclature

Much of the terminology used in data mining has grown out of terminology used in both machine learning and research statistics. We identify, for example, two very broad categories of model building algorithms as **descriptive analytics** and **predictive analytics**. In a traditional machine learning context, these equate to **unsupervised learning** and **supervised learning**. We cover both approaches in the following chapters and describe each in a little more detail in the following sections.

On top of the basic algorithm for building models, we also identify **meta learners**, which include **ensemble learners**. These approaches suggest building many models and combining them in some way. Some ideas for ensembles originate from the multiple inductive learning (MIL) algorithm (Williams, 1988), where multiple decision tree models are built and combined as a single model.

8.2 A Framework for Modelling

Building models is a common pursuit throughout life. When we think about it, we build ad hoc and informal models every day when we solve problems in our head and live our lives. Different professions, like architects and engineers, for example, specifically build models to see how things fit together, to make sure they do fit together, to see how things will work in the real world, and even to sell the idea behind the model to others. Data mining is about building models that give us insights into the world and how it works. But even more than that, our models are often useful to give us guidance in how to deal with and interact with the real world.

Building models is thus fundamental to understanding our world. We start doing it as a child and continue until death. When we build a model, whether it be with toy bricks, papier mâché, or computer software, we get a new perspective of how things fit together or interact. Once we have some basic models, we can start to get ideas about more complex ones, building on what has come before. With data mining, our models are driven by the data and thus aim to be objective. Other models might be more subjective and reflect our views of what we are modelling.

In understanding new, complex ideas, we often begin by trying to map the idea into concepts or constructs that we already know. We bring these constructs together in different ways that reflect how we understand a new, more complex idea. As we learn more about the new, complex idea, we change our model to better reflect that idea until eventually we have a model that is a good enough match to the idea.

The same is true when building models using computers. Writing any computer program is essentially about building a model. An accountant's spreadsheet is a model of something in the world. A social media application captures a model or introduces a new model of how people communicate. Models of the economy and of the environment provide insights into how these things work and allow us to explore possible future scenarios.

An important thing to remember, though, is that no model can perfectly represent the real world, except in the most simplistic and trivial of scenarios. To perfectly model the real world, even if it were possible, we would need to incorporate into the model every possible variable imaginable. The real world has so many different factors feeding into it that all we can really hope to do is to get a good approximation of it.

A model, as a good approximation of the world, will express some understanding of that world. It needs to be expressed using some language, whether it be a spoken or written human language, a mathematical language, a computer language, or a modelling language. The **language** is used to represent our knowledge.

We write or speak in **sentences** based on the language we have chosen. Some **sentences** expressed in our chosen language will capture useful knowledge. Other sentences might capture misinformation, and yet others may capture beliefs or propositions, and so on. Formally, each sentence will express or capture some concept within the formal constraints of the particular language chosen. We can think of constructing a sentence to express something about our data as building a model.

For any language, though, there is often an infinite (or at least a very large) collection of possible sentences (i.e., models) that can be expressed. We need some way of **measuring** how good a sentence is. This might just be a measure of how well formed our written sentence is—is it grammatically correct and does it read well? But just as importantly, does the sentence express a valid statement about the world? Does it provide useful insight and knowledge about the world? Is it a good model?

For each of the model builders we introduce, we will use this three-pronged framework:

> identify the **language** used to express the discovered knowledge,
>
> develop a mechanism to **search** for good sentences within the language, and
>
> define a **measure** that can be used to assess how good a sentence is.

This is a quite common framework from the artificial intelligence tradition. There we seek to automatically search for solutions to problems, within the bounds of a chosen knowledge representation language.

This framework is simply cast for the task of data mining—the task of building models. We refer to an algorithm for building a model as a **model builder**. Rattle supports a number of model builders, including clustering, association rules, decision tree induction, random forests, boosted decision trees, support vector machines, logistic regression, and neural networks. In essence, the model builders differ in how they represent the models they build (i.e., the discovered knowledge) and how they find (or search for) the best model within this representation.

In building a model, we will often look to the structure of the model itself to provide insights. In particular, we can learn much about the relationships between the input variables and the target variable (if any) from studying our models. Sometimes these observations themselves deliver benefits from the data mining project, even without actually using the models directly.

There is generally an infinite number of possible sentences (i.e., models) given any specific language. In human language, we are generally very well skilled at choosing sentences from this infinite number of possibilities to best represent what we would like to communicate. And so it needs to be with model building. The skill is to express, within the chosen language, the best sentences that capture what it is we are attempting to model.

8.3 Descriptive Analytics

Descriptive analytics is the task of providing a representation of the knowledge discovered without necessarily modelling a specific outcome. The tasks of cluster analysis, association and correlation analysis and pattern discovery, can fall under this category.

From a machine learning perspective, we might compare these algorithms to unsupervised learning. The aim of unsupervised learning is to identify patterns in the data that extend our knowledge and understanding of the world that the data reflects. There is generally no specific target variable that we are attempting to model. Instead, these approaches shed light on the patterns that emerge from the descriptive analytics.

8.4 Predictive Analytics

Often our task in data mining is to build a model that can be used to predict the occurrence of an event. The model builders will extract knowledge from historic data and represent it in such a form that we can apply the resulting model to new situations. We refer to this as predictive analytics.

The tasks of classification and regression are at the heart of what we often think of as data mining and specifically predictive analytics. Indeed, we call much of what we do in data mining predictive analytics.

From a machine learning perspective, this is also referred to as supervised learning. The historic data from which we build our models will already have associated with it specific outcomes. For example, each observation of the *weather* dataset has associated with it a known outcome, recorded as the target variable. The target variable is `RainTomorrow` (whether it rained the following day), with the possible values of `No and Yes`.

Classification models are used to predict the class of new observations. New observations are classified into the different target variable categories or classes (for the *weather* dataset, this would be Yes and No). Often we will be presented with just two classes, but it could be more. A new observation might be today's weather observation. We want to classify the observation into the class Yes or the class No. Membership in a particular class indicates whether there might be rain on the following day or not, as the case may be.

Often, classification models are represented symbolically. That is, they are often expressed as, for example, a series of tests (or conditions) on different variables. Each test exhibits a piece of the knowledge that, together with other tests, leads to the identified outcome.

Regression models, on the other hand, are generally models that predict a numeric outcome. For the *weather* dataset, this might be the amount of rain expected on the following day rather than whether it will or won't rain. Regression models are often expressed as a mathematical formula that captures the relationship between a collection of input variables and the numeric target variable. This formula can then be applied to new observations to predict a numeric outcome.

Interestingly, *regression* comes from the word "regress," which means to move backwards. It was used by Galton (1885) in the context of techniques for regressing (i.e., moving from) observations to the average. The early research included investigations that separated people into different classes based on their characteristics. The regression came from modelling the heights of related people (Crano and Brewer, 2002).

8.5 Model Builders

Each of the following chapters describes a particular class of model builders using specific algorithms. For each model builder, we identify the structure of the language used to describe a model. The search algorithm is described as well as any measures used to assist in the search and to identify a good model.

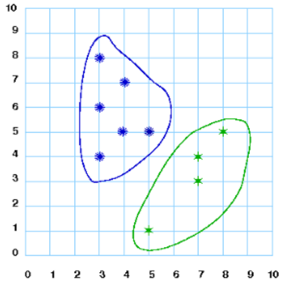
Following the formal overview of each model builder, we then describe how the algorithm is used in Rattle and R and provide illustrative examples. The aim is to provide insight into how the algorithm works and some details related to it so that as a data miner we can make effective use of the model builder.

The algorithms we present will generally be in the context of a two-class classification task where appropriate. The aim of such tasks is to distinguish between two classes of observations. Such problems abound. The two classes might, for example, identify whether or not it is predicted to rain tomorrow (`No` and `Yes`). Or they might distinguish between high-risk and low-risk insurance clients, productive and unproductive taxation audits, responsive and nonresponsive customers, successful and unsuccessful security breaches, and so on. Many of the popular algorithms are covered in the following chapters. Algorithms not covered include neural networks, linear and logistic regressions, and Bayesian approaches.

In demonstrating the tasks using Rattle (together with a guide to the underlying R code), we note that Rattle presents a basic collection of tuning parameters. Good default values for various options allow the user to more simply build a model with little tuning. However, this may not always be the right approach, and whilst it is certainly a good place to start, experienced users will want to make much more use of the fuller set of tuning parameters available directly through the R Console.

Chapter 9

Cluster Analysis



The clustering technique is one of the core tools that is used by the data miner. Clustering gives us the opportunity to group observations in a generally unguided fashion according to how similar they are. This is done on the basis of a measure of the distance between observations. For example, we might have a dataset that is made up of school children of various heights, a range of weights, and different ages. Depending on what is needed to solve the problem at hand, we might wish to group the students into smaller, more definable groups and then compare different variables common to all groupings. Each group may have different ranges, minimums and maximums, and so on that represent that group. Clustering allows the data miner to break data into more meaningful groups and then contrast the different clusters against each other. Clusters can also be useful in grouping observations to help make the smaller datasets easier to manage. The aim of clustering is often to identify groups of observations that are close together but as a group are quite separate from other groups.

Numerous algorithms have been developed for clustering. In this chapter, we focus primarily on the k-means clustering algorithm. The algorithm will identify a collection of k clusters using a heuristic search starting with a selection of k randomly chosen clusters.

G. Williams, *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*, Use R, DOI 10.1007/978-1-4419-9890-3_9, © Springer Science+Business Media, LLC 2011

9.1 Knowledge Representation

A model built using the k-means algorithm represents the clusters as a collection of k means. The observations in the dataset are associated with their closest “mean” and thus are partitioned into k clusters. The *mean* of a particular *numeric* variable for a collection of observations is the average value of that variable over those observations. The *means* for the collection of observations that form one of the k clusters in any particular clustering are then the collection of mean values for each of the input variables over the observations within the clustering.

Consider, for example, a simple and small random subset of the *weather* dataset. This can be generated as below, where we choose only a small number of the available numeric variables:

```
> library(rattle)
> set.seed(42)
> obs1 <- sample(1:nrow(weather), 5)
> vars <- c("MinTemp", "MaxTemp",
            "Rainfall", "Evaporation")
> cluster1 <- weather[obs1, vars]
```

We now obtain the means of each of the variables. The vector of means then represents one of the clusters within our set of k clusters:

```
> mean(cluster1)

    MinTemp     MaxTemp    Rainfall Evaporation
       4.74       15.86        3.16        3.56
```

Another cluster will have a different mean:

```
> obs2 <- setdiff(sample(1:nrow(weather), 20), obs1)
> cluster2 <- weather[obs2, vars]
> mean(cluster2)

    MinTemp     MaxTemp    Rainfall Evaporation
     6.6474     19.7579      0.8421      4.4105
```

In comparing the two clusters, we might suggest that the second cluster generally has warmer days with less rainfall. However, without having actually built the clustering model, we can’t really make too many such general observations without knowing the actual distribution of the observations.