

THE INSIDERS' GUIDE TO **PREDICTIVE ANALYTICS**



Jarlath Quinn

Smart Vision Europe

THE INSIDERS' GUIDE TO PREDICTIVE ANALYTICS

Jarlath Quinn

Smart Vision Europe

Copyright © 2020 Jarlath Quinn.

All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher except for the use of brief quotations in a book review.

ISBN: 978-1-8380581-0-4 (Paperback)

ISBN: 978-1-8380581-1-1 (eBook)

Book design by Starfish Limited, Norwich.

Printed by Amazon in the UK.

First printing edition 2020.

Smart Vision Europe Limited
Burlingham House
Norwich Road
Saxlingham Nethergate
Norwich
NR15 1TP

www.sv-europe.com

TABLE OF CONTENTS

Table of contents	5-6
Table of figures	7
Introduction	9
Defining predictive analytics	10
'Tell us something we don't know'	13
Chapter 1 Choosing a predictive analytics project	16
1.1 Finding inspiration	16
1.2 The CRISP-DM model	18
1.2.1 Business understanding	22
1.2.2 Making a plan	27
Chapter 2 The raw material	29
2.1 Data understanding	29
Chapter 3 Shaping the data	35
3.1 Data preparation	35
3.1.1 Merging and appending files	38
3.1.2 Aggregating data	38
3.1.3 Transposing data	39
3.1.4 Creating new fields	39
Chapter 4 The algorithm menagerie	41
4.1 Predictive models	41
4.2 Segmentation models	42
4.3 Association models	43
4.4 Other model types	44
4.5 The two cultures	44
4.6 Statistical techniques	47
4.7 Rule induction / decision trees	49
4.8 Machine learning	52

Chapter 5	Building a predictive model	55
Chapter 6	What does ‘good’ look like?	74
6.1	Accuracy	76
6.2	Interpretability	77
6.3	Stability	77
6.4	Coherence	77
6.5	Simplicity	78
6.6	Performance	78
6.7	Visualising model performance	81
6.8	Model performance metrics	85
6.8.1	Overall accuracy	86
6.8.2	Area under the curve	86
6.8.3	Gini coefficient	86
6.8.4	Lift	86
6.9	Model validation	87
6.9.1	Training / testing sample split	87
6.9.2	Cross-validation	88
Chapter 7	Back in the real world	91
7.1	Creating selections	92
7.2	Testing deployment	98
Chapter 8	Beyond deployment	103
8.1	Monitoring performance	103
8.2	Automation	103
8.3	Planning and deciding	104
8.4	Last thoughts	105
Chapter 9	Bibliography	107

TABLE OF FIGURES

Figure 1 - Scored customer table showing propensity value, cluster group membership and anomaly index

Figure 2 - The CRISP-DM process

Figure 3 - Typical team roles in a predictive analytics project

Figure 4 - Breakdown of the business understanding phase into tasks and sub-tasks

Figure 5 - A sample of possible fields that might be utilised to predict contract renewal

Figure 6 - Breakdown of the data understanding phase into tasks and sub-tasks

Figure 7 - Report output showing summary measures for a data file with around 89,000 records

Figure 8 - Exploring the relationship between the customer age group and customer churn

Figure 9 - The two-way relationship between business understanding and data understanding

Figure 10 - Breakdown of the data preparation phase into tasks and sub-tasks

Figure 11 - The two-way relationship between data preparation and modelling

Figure 12 - Statistical terms and their machine learning equivalents

Figure 13 - Scatterplot showing relationship between horsepower and engine size for a sample of cars

Figure 14 - Example decision tree using the CHAID algorithm to predict survival on the RMS Titanic

Figure 15 - Decision tree rules showing predictions for the two groups with the highest and lowest chance of survival respectively on the RMS Titanic

Figure 16 - The three main families of analytical approaches with example algorithms

Figure 17 - Breakdown of the modelling phase into tasks and sub-tasks

Figure 18 - Selection of fields from a prepared data file used to build a model predicting customer churn

Figure 19 - The root node of the CHAID interactive decision tree

Figure 20 - Potential predictor variables that may be entered into the CHAID decision tree model ordered by statistical significance

Figure 21 - The first branch of the interactive CHAID decision tree split by the variable Auto_Renew

Figure 22 - Potential predictors that may be entered under the 'Month-to-month' node in the decision tree

Figure 23 - The CHAID interactive decision tree grown two levels deep via the 'Month-to-month' contract branch

Figure 24 - The partially built decision tree model applied to a validation tree based on a withheld 30% random subset of data

Figure 25 - The CHAID interactive decision tree grown three levels deep via the Month-to-month contract and DSL connection branch

Figure 26 - The partially built decision tree model grown three levels deep but displayed in the model validation tree

Figure 27 - The complete CHAID decision tree based on the entire training sample of 3,359 records

Figure 28 - Predictor importance chart showing the relative impact of the included decision tree variables on model accuracy (larger bars indicate greater importance)

Figure 29 - Coefficients and model terms table generated by running a logistic regression procedure

Figure 30 - Predictor importance chart showing the relative impact of the included logistic regression variables on model accuracy

Figure 31 - Output diagram showing a Neural Network model with a single neuron in a hidden layer

Figure 32 - Output diagram showing a Neural Network model with multiple neurons in a hidden layer

Figure 33 - Predictor importance chart showing the relative impact of the Neural Network variables on model accuracy

Figure 34 - The CRISP-DM methodology showing a direct link between the evaluation phase and the business understanding phase

Figure 35 - Breakdown of the evaluation phase into tasks and sub-tasks

Figure 36 - Comparing the accuracy of three models using tables

Figure 37 - Gains chart showing model performance compared to a random and a 'perfect' model

Figure 38 - Gains chart showing the proportion of churners detected by selecting the top 27% of the data in terms of the model confidence

Figure 39 - Gains chart comparing performance for two predictive models

Figure 40 - ROC curve showing the model performance in terms of the relationship between true positives and false positives

Figure 41 - Comparison of model performance between training and testing samples (test group based on a random 30% sample of cases)

Figure 42 - Illustration of 10-Fold cross-validation procedure

Figure 43 - Breakdown of the deployment phase into tasks and sub-tasks

Figure 44 - A scored data file containing Customer ID values and model prediction fields that indicate each customer's likelihood of churning

Figure 45 - Histogram of churn probability scores generated by a model

Figure 46 - Selecting circa 20,000 customers with the highest risk of churn

Figure 47 - Sample list of customers with a churn risk of at least 0.4 (40%)

Figure 48 - Profit chart showing estimated campaign profitability based on expected revenue of £62 for successfully retaining a customer and expected offer costs of £20

Figure 49 - Selecting the 36% of customers with the highest probability of being retained generates the largest estimated overall profit for the retention campaign

Figure 50 - Creating test groups to establish model performance in the real world

Figure 51 - The outer circle of the CRISP-DM diagram indicating the iterative nature of the process model

INTRODUCTION

Sometime around the start of this century, a new phrase began to emerge describing the exciting kinds of advanced analytics that organisations with lots of data could exploit. As early as 2001 (Krill, 2001) authors began to make reference to technology vendors who could provide ‘predictive analytics’. These vendors supplied software that enabled their customers to go far beyond the typical analytical functions of reporting key metrics, analysing survey responses or examining research data.

Working with a mix of statistical, data mining and machine learning approaches, they were developing sophisticated solutions that used analytics to address a wide range of problems - anything from predicting insurance fraud or identifying businesses that misstated their tax obligations, to estimating when manufacturing equipment required maintenance or recommending the most appropriate product to offer retail customers.

In early 2003, the corporate marketing department at the analytics tech company SPSS Inc decided to embrace predictive analytics as the centrepiece of its go-to-market strategy. I had been working for SPSS in the UK since 1995 and I confess that my initial reaction to this development was one of confusion. Why did we need a new term to describe our offerings? We already had a portfolio of technology devoted to disciplines such as classical statistics, data mining and text analytics. Why narrow it down to ‘predictive’ analytics when so much of what we did wasn’t necessarily focussed on predicting things?

But then I didn’t work in marketing. My role was to support the field sales team when they discussed software solutions with potential customers. This often entailed talking about the more esoteric aspects of data analysis with statisticians, researchers, academics, engineers and IT experts. However, as a Sales Engineer (or SE), I also met with customers and prospects who did *not* have an analytically technical background and, more importantly, who tended to have a stronger influence on software procurement.

I was required to explain the benefits of these technologies whilst avoiding boring my audience with the finer points of difference between, say, descriptive statistics and machine learning. It was a balancing act and my colleagues and I were always careful to avoid accusations of ‘dumbing things down’ by our more technical peers.

The problem was that in most meetings with prospective customers, too much time was spent explaining how our analytical capabilities differed from the functionality they already had. It became clear that the prospects often didn’t seem to really understand what it was that we were selling.

This became very clear when we asked them which of our competitors they were also considering. A surprising amount of time they mentioned companies like Business Objects or Cognos. These were business intelligence (BI) vendors that focussed on sophisticated executive dashboards and interactive reporting technology, not data

mining or advanced statistics. Clearly someone was in the wrong meeting.

The issue was compounded by the fact that it's very hard for sales staff in emerging technology markets to turn down an opportunity to present to a prospective customer, even when that customer is looking for a completely different set of capabilities or may have already chosen their preferred supplier and doesn't understand what you're offering in the first place. I recall once pressing a colleague to explain to me exactly what the client expected to see and which of our solutions we should lead with, only to be told by the exasperated sales exec, 'Just show them something whizzy!'

It took a while for me to understand that predictive analytics is really a catch-all phrase. It's an umbrella term that covers a lot of individual capabilities under a single heading, rather than a precise definition of particular techniques. Ultimately, I realised that this didn't matter. After all, business intelligence was a very successful umbrella term that, despite its deliberate vagueness, was widely understood by the market. It seems that technology markets like these powerful-sounding descriptions, and in that sense, predictive analytics worked.

Defining predictive analytics

So, what is predictive analytics? Back in January 2003, SPSS Inc chose the following definition:

'Predictive analytics connects data to effective action by drawing reliable conclusions about current conditions and future events.'

You may notice a couple of things here. Firstly, there's no mention of technology or terminology such as 'statistical' or 'machine learning'. It could be referring to any kind of analytical approach. Secondly, there's an emphasis on connecting data to *action*, rather than something like 'insight'.

At the time, I liked the fact that they tried to take the technical sting out of the definition and that they put decision-making (i.e. action) at the forefront. But it's not a particularly memorable definition, and that last part about 'drawing reliable conclusions about current events' seems a little clunky. Surely any kind of data analysis is about drawing 'reliable conclusions' and isn't BI technology at least optimised for tracking 'current conditions'?

Nevertheless, clients suddenly seemed to understand that this represented a distinct and powerful use of analytics. So much so that this shift in positioning helped SPSS to steadily increase its revenue by 50% over 5 years.

In 2009 IBM acquired SPSS Inc for \$1.2 billion and immediately became one of the leading vendors of predictive analytics technology. IBM's own definition (at time of writing) is somewhat more verbose:

'Predictive analytics brings together advanced analytics capabilities spanning ad-hoc statistical analysis, predictive modeling, data mining, text analytics, optimization, real-time scoring and machine learning.'

These tools help organisations discover patterns in data and go beyond knowing what has happened, to anticipating what is likely to happen next.’ (Anon., 2020)

Wikipedia defines predictive analytics thus:

‘Predictive analytics encompasses a variety of statistical techniques from data mining, predictive modelling, and machine learning, that analyse current and historical facts to make predictions about future or otherwise unknown events.’ (Anon., 2020)

Neither of these more recent definitions can be said to easily trip off the tongue. Indeed, they effectively state the same thing: predictive analytics uses 1) a bunch of advanced analytical approaches to 2) find patterns in data and make predictions.

Perhaps trying to come up with a single, snappy definition misses the point. After all, this was a phrase originally invented to make it easier to sell software that might otherwise have been confused with BI reporting platforms or confined to the technical worlds of computer science and academia.

Nevertheless, it *is* true to say that the kinds of techniques involved in predictive analytics are fairly wide-ranging. They can constitute everything from bleeding-edge, deep learning methods to statistical approaches that *pre-date* the advent of modern computing. Later in this book we’ll take a closer look at the menagerie of algorithms that predictive analytics encompasses. But for now, the more important point is that these techniques all share one common capability: they can create new data.

It’s worth taking a moment to consider the implications of this seemingly inconsequential technical detail. Figure 1 shows a sample from a data file that has been used in a predictive analytics project.

	ACNO	Response Propensity	Cluster Group	Anomaly Index
197	617640	0.053	cluster-5	0.70
198	618322	0.001	cluster-2	0.98
199	618332	0.057	cluster-5	1.06
200	618522	0.104	cluster-5	5.92
201	618718	0.028	cluster-5	0.78
202	618841	0.001	cluster-3	1.49
203	618922	0.002	cluster-5	1.20
204	618931	0.490	cluster-5	0.92
205	619035	0.500	cluster-3	0.91
206	619056	0.004	cluster-5	0.80
207	619189	0.057	cluster-5	0.76
208	619208	0.037	cluster-5	1.47
209	619252	0.104	cluster-5	0.76
210	619984	0.104	cluster-5	1.34
211	620389	0.001	cluster-3	0.85
212	620650	0.036	cluster-5	1.17
213	621134	0.036	cluster-5	0.76
214	621189	0.002	cluster-3	0.94
215	621560	0.001	cluster-1	0.82
216	625562	0.036	cluster-5	0.91

Figure 1 - Scored customer table showing propensity value, cluster group membership and anomaly index

You can see that the rows in the file represent different customer accounts (the column labelled 'ACNO' refers to the account number). You can also see that there are three additional fields: 'Response_Propensity', 'Cluster_Group' and 'Anomaly_Index'. Each one of these fields has been *generated* using a separate analytical algorithm. We don't know what data the algorithms analysed to create these columns. Perhaps they used customers' demographic details such as gender and age, or additional financial information such as average amount spent or the elapsed time since the most recent transaction, or details about how they interacted with the business whether via a website, a mobile app or in person.

The process of creating fields based on the algorithmic analysis of historical data is called 'scoring' and this is an example of a data extract from a 'scored' customer table. In this instance, it doesn't matter what kind of organisation these data come from. Later, we will take a more detailed look at how this process occurs, but it's important to understand that this is what predictive analytics is about. It creates new data in order to help people make better decisions or take the most appropriate action.

Images like Figure 1 aren't usually shown in adverts and articles about predictive analytics (or machine learning, or data science or AI - delete as appropriate). Instead

we tend to see either abstract depictions representing algorithms, surging oceans of data, illuminated human brains or exciting screen grabs of complex and colourful data visualisations. In contrast, pictures like Figure 1 just don't look that compelling. But of course, the kind of data we're being shown in this image represent the absolute fulcrum of analytical power. That's because the physical output of a predictive analytics application is usually a column, or series of columns, created with the sole purpose of helping us to make smarter choices. Depending on the aims of the application itself, these new data could take many forms. For example:

- A series of probabilities showing the likelihood that different customers will renew their subscriptions
- A cluster membership field indicating the segments or natural groupings that people belong to, based on their transactional, demographic and behavioural data
- A column of risk scores between 1 and 100 indicating the chance that different machines or assets will need maintenance within the next few days
- A field showing anticipated exam results for students based on their past performance
- A numerical index indicating the degree to which individual financial transactions are anomalous
- A series of fields that categorise the sentiments of guests based on their written responses on a hotel review app
- A group of variables showing which products are most appropriate to offer customers the next time they visit a website

'Tell us something we don't know'

It helps to keep in mind that whilst these newly created data values are often the initial prize of a predictive analytics application, they don't represent the final goal. For the moment, that could loosely be expressed as 'doing things differently'. It's also important to understand that, with predictive analytics applications, our future decisions are calibrated by the values that the application generates.

For instance, a common assumption is that predictive analytics will always generate deeper insights for the business. In fact, that isn't necessarily the case. Working as a sales engineer, my job involved explaining how particular analytical solutions worked, demonstrating software, answering technical questions and, crucially, conducting proof-of-concept projects using the prospect's own data.

Proof-of-concept projects are especially tricky because often the client only has a vague notion of how they will evaluate the results. Furthermore, the projects themselves usually have very limited resources in terms of both support and

available time (typically 4 or 5 days to complete). When it came to proof-of-concepts however, for both myself and my colleagues, by far the *single worst brief* was ‘Tell us something we don’t know’.

A brief like this indicates that the prospective client thinks predictive analytics represents a sort of super-charged version of business intelligence, that this approach will whisper some deep, but hitherto unknown, truth in their ear concerning a critical aspect of the business. Worse still, often the prospect is unwilling or unable to stipulate what useful things they don’t know. So, the brief becomes ‘Tell us something we don’t know... but we *ought* to know’. The obvious answer to this of course, is that they ought to know they’re completely missing the point.

Although predictive analytics is focused on developing applications to aid decision-making, the journey towards this goal sometimes yields powerful insights as a by-product of the process. But it’s not a guarantee. If, for example, a company is trying to predict the likelihood of customers renewing their contracts, they may already know that this is affected by gender or age group or product category. They shouldn’t be surprised therefore to find that those factors are the main components driving the predictive model.

What predictive analytics does well is uncover combinations of those same factors in order to generate a value that correlates with the target outcome in question – in this case, the likelihood to renew a contract. I found myself in exactly this situation, when presenting the results of a predictive model and one of the key stakeholders complained that ‘it’s not telling us anything new’.

The model in question happened to use a number of variables that they already understood could affect the outcome. But the model also accurately quantified the risk of the critical event occurring. This was something the organisation had not been able to do previously. Presumably, the stakeholder felt that the purpose of the project should have been to uncover important new factors rather than simply predict the future. They were evaluating the application in terms of any new insight it might deliver rather than its predictive power.

In fact, depending on the application, it might turn out that the most accurate technique uses what’s known as a ‘black box’ method where the components of the predictive model are effectively hidden – you don’t know what combination of variables are driving the model. In these cases, any potential insight is willingly sacrificed for increased accuracy.

This point touches on a fundamental aspect of the process of planning and developing new predictive analytics applications – determining what success looks like. For example, what criteria should we use? Is accuracy more important than insight? What do we regard as ‘accurate’ anyway? As we shall see, these questions are best addressed before we start to crunch data as they’re very important in terms of the ultimate success or failure of the project.

Speaking of failure, where are all the news stories and journal articles about advanced analytics projects that were abandoned? At the time of writing, a quick web search revealed a torrent of success stories including an article about how machine learning is used to identify hidden geological features such as cave entrances (Geological Society of America, 2019), a new system that estimates the risk of cardiovascular death (Gordon, 2019), an AI platform that can predict flu outbreaks with 90% accuracy (Goscha, 2019) and a predictive analytics model that prophesised New Zealand would beat Wales in the final of the 2019 Rugby World Cup (Daniel, 2019)¹.

Not all stories shine a positive light on advanced analytics. It's also easy to find plenty of articles and editorials on the recurrent themes of 'the dangers of AI' or the 'limitations of machine learning', but it's a little harder to find documented case histories concerning predictive analytics projects that either simply didn't work, or that for one reason or another, failed to make an impact in the real world.

In reality, as we would expect with any innovative field, there's no shortage of these events. In this book, I intend to draw upon my own experiences and those of my colleagues to explore precisely what drives success or failure in predictive analytics projects. When it works, why does it work and what can be done to manage the risks?

1 The final of the 2019 Rugby World Cup was played between England and South Africa. South Africa won.

CHAPTER 1

CHOOSING A PREDICTIVE ANALYTICS PROJECT

1.1 Finding inspiration

Given that there are so many different areas where predictive analytics can be used, how do you select which problems you want to focus on in a project?

Much of this has to do with complexity and value. A key property of apparent complexity is variation. When there's no simple, single rule to explain the variation we observe in our everyday lives (such as the time it takes to get to work, how many days of sick leave we will need, the number customer complaints we receive or the office Wi-Fi availability) we tend to see complexity.

Empirical analysis of data aims to make sense of a complicated thing by trying to account for its variation. Indeed, statistics as a discipline could be simply described as the 'the science of variation.' Most organisations are either already complex or they engender complexity, meaning that there are plenty of examples of outcomes that vary.

However, a lot of this variation is either something that isn't regarded as terribly important, such as the number of glasses of water people in a department drink or who gets the last parking space, or the variation itself can't realistically be accounted for, such as the names of people who will become customers next year or the company stock price at the end of the next quarter.

Therefore, any useful predictive analytics initiative needs to focus on subject areas that meet two basic criteria: firstly, they are regarded as valuable and secondly, there is a reasonable chance that the variation can be accounted for.

But there are other things that we need to consider as well. What if we identify a subject area that is clearly valuable and where we may be able to account for the variation (i.e. it's potentially predictable) but where there's little we can do to either influence the outcome or alter its effects? If you work for an airline, you may well be able to estimate the effect of a last-minute flight cancellation on your passengers' satisfaction levels but being able to predict this accurately doesn't mean there's much you can do to prevent it upsetting them.

Then there's the timeliness of predictions to consider. Being able to forecast whether or not it's going to rain with 99% accuracy might sound impressive, but less so if it can only be done one minute before the first raindrops start to appear. Similarly, you might be able to accurately identify mobile phone users who are likely to cancel their contracts one month before they expire, only to find that unfortunately, that's too late. You can't do much to change their minds without incurring unacceptable costs. These last examples relate to a third criterion: the outcome of the predictive analytics application should be actionable.

Bearing in mind that the most successful initiatives of this kind focus on areas which

are valuable, where the outcome in question is predictable and where the results are actionable, perhaps we shouldn't be surprised that, despite the constant torrent of news articles regarding the latest developments in machine learning, data science or AI, the authors often overlook one salient truth: most advanced analytics projects fail because they never even get started. Often this is simply because people don't know what they should focus on first or what is most likely to gain the support of senior management. The good news is that sometimes the most obvious targets for these initiatives are hiding in plain sight.

Consider an organisation's strategic objectives. Clearly, these are valuable and are prone to unwanted variation if not addressed properly, so much so that they are publicly stated as yearly goals in the annual report. At the time of writing, a random web search of annual reports yielded strategic objectives such as 'deepening customer engagement' (Vodafone, 2019), driving 'data to insight to action' (Thames Water, 2019), delivering a 'personalized guest experience' (Hilton, 2018) and 'bearing down on avoidance and evasion' (HMRC, 2019).

A predictive analytics application could play a role in addressing all these objectives. Furthermore, someone looking for internal sponsorship to support an analytics initiative might have a significantly stronger case if the project directly addresses a strategic level objective.

Strategic objectives often have a necessary vagueness to them that suggest more than one specific target. Drilling down on Vodafone's 2019 objective of 'deepening customer engagement' it becomes apparent that this refers to customer retention and 'upselling converged offers and additional services'. Both of these areas are classic predictive analytics targets in the telecoms sector.

There are also plenty of potential targets for advanced analytics to be found at a more tactical level. Think about what is being monitored in the organisation. The most obvious examples of this are shown in KPIs (key performance indicators). In fact, many executives (and potential sponsors) are directly compensated on performance metrics related to customer loyalty and contentment such as Net Promoter Score™ (NPS) or customer satisfaction metrics (CSAT). Meanwhile entire departments are sometimes evaluated against measures such as average revenue per user (ARPU), payment error rate, quote-to-close ratio, median time to resolution, student retention rate, percentage out of stock items, and average waiting times.

These are outcomes that have already been quantified, that often have a recorded history, that are regarded as valuable and are clearly prone to variation. Most importantly these tactical measures, like strategic goals, are meaningful to the business rather than merely interesting to the data analyst.

So far, we've discussed potential areas that can provide inspiration for predictive analytics applications. In these contexts, new analytical initiatives may be seen as innovative or ground-breaking. Sometimes however, the initiative has already been seized by someone else and the main driver of the project is really a need to respond

to that.

In his excellent book *Guns, Germs and Steel* (Diamond, 1997), Jarrod Diamond argues that, contrary to popular belief, necessity is not the mother of invention - in fact the reverse is more usually true. Diamond painstakingly illustrates how many major technological breakthroughs, ranging from the airplane and automobile, through to the internal combustion engine and the lightbulb, began as inventions in search of a use, solutions in search of a problem.

This reminded me of an occasion when I was asked to deliver a training course in the use of predictive analytics software for a global engineering company. It was an unusual situation because there was only one training attendee, a senior executive with limited hands-on experience of data analysis.

When I asked her why she chose this particular course, she explained that her company's largest competitor had recently trumpeted its successful deployment of predictive analytics to retain key customers and improve its services. As a result of this, she wished to understand how similar technology could be used in her own organisation. It was this innovation, made by a competitor, that had caused her to regard predictive analytics as an imperative.

It's an uncomfortable position for any commercial organisation to find itself having to play catch-up with a competitor, but as stated earlier, given that most advanced analytics projects fail because they never even get started, there's something to be said in favour of innovation for innovation's sake.

1.2 The CRISP-DM model

Throughout this book we will examine the factors that encourage success or increase the chances of failure in predictive analytics applications. We began this chapter by discussing what might inspire a new analytics project in the first place. But even when one successfully identifies a compelling subject or target area for a predictive analytics project, there's no guarantee that the results will be judged a success.

By 'success' I mean that the application drove outcomes that were demonstrably better than previous approaches. This being the case, if there is one thing that increases the likelihood of a favourable result it is effective planning. Of course, any project or initiative is more likely to succeed if it is well planned from the start. But the problem with predictive analytics in particular is that organisations and individuals with little experience of developing new applications in this area tend to focus on the analytical part of the solution, at the expense of the context that it exists within. Remember that the purpose of the analytics is to suggest new courses of action. So thinking about what these new actions might entail, how they will be evaluated and who will be impacted by them should be figured out early on in the project cycle.

Unlike exploratory analysis or data visualisation exercises, predictive analytics focuses on changing behaviour rather than merely gaining insights into a particular phenomenon.

Fortunately, some of this planning is made easier by the availability of existing methodologies that are specifically designed to help organisations develop analytical applications. Some of these methodologies are domain-specific or designed to work with particular software tools, but to date the most popular open-standard methodology is CRISP-DM.

CRISP-DM stands for Cross Industry Standard Process for Data Mining. The methodology was first conceived in 1996 when the term ‘data mining’ was widely used to describe the kinds of applications that were later associated with the wider scope of predictive analytics (Shearer, 2000). The upshot of this is that we can just as easily regard it as the Cross Industry Standard Process for Predictive Analytics (CRISP-PA?).

A methodology like CRISP-DM is useful for two main reasons. Firstly, it outlines the process of developing a predictive analytics application in a fairly transparent and logical fashion. This is especially useful for any non-technical stakeholders involved the project. Secondly, as a process model it provides a series of check points that those involved in the project are compelled to address. This means that it lessens the chance that some vital detail that could scupper the project is overlooked.

Figure 2 shows a classic depiction of the CRISP-DM model as a circular process. This simply reflects the fact that developing a predictive analytics application is usually an iterative undertaking. As you can see, the methodology is broken down into six main phases and we’ll go on to discuss each of these in turn throughout this book.

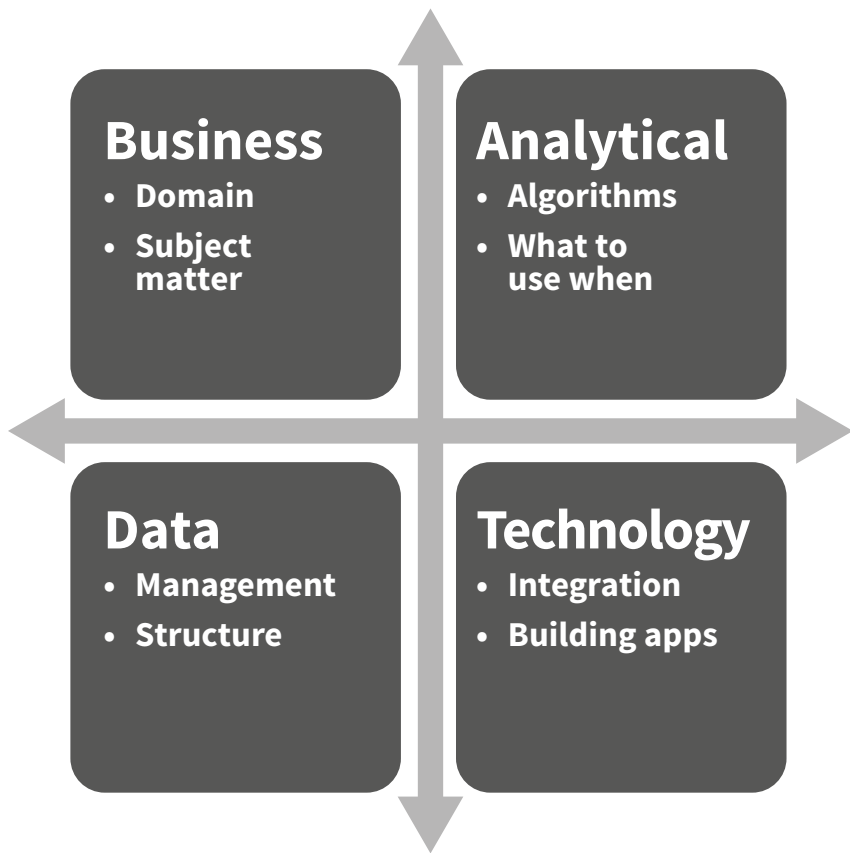


Figure 3 - Typical team roles in a predictive analytics project

To begin with, the person responsible for a specific organisational concern or line-of-business such as customer loyalty, student satisfaction, outbound marketing, asset maintenance, patient safety, supply chain or tenant arrears will almost certainly have extensive knowledge of the context that the application will be applied in. Their input is likely to be of critical importance.

Similarly, someone who manages the organisation's data resources is probably well placed to advise the project stakeholders of the reliability of the data or be able to provide access to key data sources such as call logs, social media posts or website traffic that may help to enhance the accuracy of the outputs and provide valuable insight.

Lastly, the project may require the assistance of personnel who can help with deploying the output of the application in the form of alerts, recommendations,

next-best-offers, estimated risk scores or propensity values to internal operational systems or customer interfaces.

At Smart Vision Europe, we're regularly asked what sort of people are needed to make these kinds of projects successful. Interestingly, we've repeatedly found that the most effective trailblazers of predictive analytics are not necessarily statistically proficient or skilled in data science, but they do tend to be data literate and business focussed. By 'data literate' we mean that they are used to dealing with raw data files, working with spreadsheets, digesting reports and explaining the context in which the data are collected. By 'business focussed' we mean that they understand what outcomes from the predictive analytics application the organisation is most likely to value, as well as the changes they might make and how they would measure the impact of those.

These kinds of people are often surprisingly adept at learning how to prepare data, develop a model, assess its performance and deploy the results. Knowing how to do this, as well as understanding how to apply these applications to drive better decision-making, means that they can become very effective pioneers of advanced analytics within their own organisations.

Because the context in which these new applications are developed is so critically important, the advice we give when projects are in their early stages is to focus most effort on the first and last phases of CRISP-DM, that is on the business understanding and deployment phases. That's because these two phases compel the project team to address the following key questions: 'Why did you choose this target?' and 'What will you do differently?'

CRISP-DM is such a valuable lens through which to view the arc of building a predictive analytics application that we can use it as a template to structure the chapters in this book. Each phase in the process allows us to explore many important aspects of the project work and the questions we need to answer in order to deliver a successful outcome. With that in mind, let's take a look at the first critical phase in the methodology - business understanding.

1.2.1 Business understanding

One of the tougher realities of delivering solutions based on predictive analytics is that you inevitably end up requiring the client to work a little harder than they're used to doing with analytical initiatives. In other projects the goal might be to prepare the data so that it can be effectively surfaced via a dashboard or explored through an exciting visualisation platform. Alternatively, the focus might be more insight-driven, where the task is to address a series of hypotheses such as 'Do promotions drive increased customer loyalty?' or 'Are profitable customers demographically different from unprofitable ones?'

In the context of a predictive analytics project, however, we're normally trying to generate new information based on historical data that will enable us to take a

more effective action. To do this well, we will need to build a very solid foundation of rationale for the project and figure out what a successful outcome would look like. Once you start to scratch the surface of this task, you often find out that there are quite a lot of factors to consider.

The CRISP-DM model stipulates that the business understanding phase has 4 primary tasks:

1. Determine the business objectives
2. Assess the situation
3. Determine the analytical goals
4. Produce a project plan

As Figure 4 shows, each step is then further broken down into a series of generic tasks, specialised tasks and process instances.

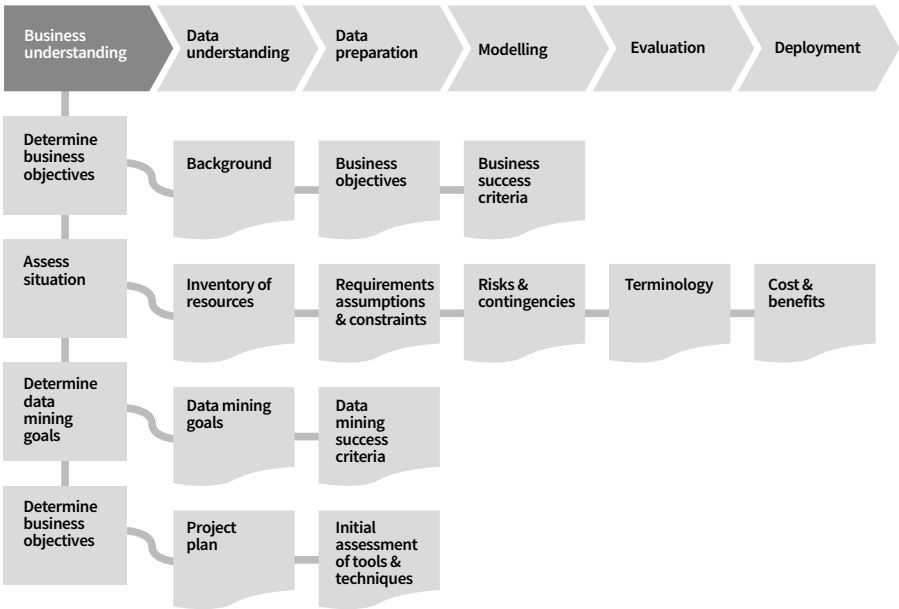


Figure 4 - Breakdown of the business understanding phase into tasks and sub-tasks

To the newly initiated, this might all seem a tad dogmatic. After all, to a certain extent this is a common-sense approach that could be applied to any project. Nevertheless, I never cease to be amazed at how often projects are initiated when even the business objectives have not been clearly stipulated. As the authors of CRISP-DM point out: 'A possible consequence of neglecting this step is to expend a great deal of effort producing the right answers to the wrong questions.'

For our purposes, let's assume that the primary business objective for our project is increasing customer loyalty. When we talk about determining the business objectives, the first question we are addressing is 'why this particular subject?' The answer might be that it meets several criteria, such as:

- Increasing customer loyalty has been stipulated as a strategic goal for the company. It has been recognised that costs associated with customer acquisition have been rising in recent years and that increasing pressure from competitor activity has led to a slowdown in the market share growth. This is therefore regarded as a valuable objective.
- Due to the contractual basis of our business model, we can identify precisely when new customers are acquired and when they cease to transact with us. That means that this is a measurable outcome.
- Previous analysis using satisfaction surveys has shown that if we are able to identify customers who are likely to cancel their contracts three months before their termination date, we can persuade around 50% of them to renew their contracts with us. Thus this can also be regarded as an actionable outcome.

The example deals with a commercial organisation attempting to maintain and grow its market share. In this case, it's not practical to get every customer to complete a satisfaction survey three months before their termination date, so the organisation needs to be able to predict the likelihood that each customer will renew their contract.

CRISP-DM dictates that a key output from the task of 'Assessing your situation' should be to figure out the rationale for the application in terms of the financial consequences. We only need to employ some rough estimates to lay bare the costs and benefits of retaining existing customers as opposed to trying to find new ones.

- The average cost incurred with persuading customers to renew an existing contract is £35 (let's assume this is only incurred by those who agree to renew)
- The average annual revenue received from a contract is £132
- The cost per head of acquiring and onboarding *new* customers is £45

To work out the impact of these estimates on our project, it helps if we do some basic calculations. If we assume that it only takes one month to replace a customer, then the lost revenue on average is only £11. However, the acquisition and onboarding costs of replacing a customer mean that this value jumps to £56. If the company is losing 100,000 customers a year (a not unreasonable number) then the total costs are £5.6 million.

Now we've established the rough costs, we need to imagine what a predictive model might do to help. Let's assume a modest-performing model identifies what it thinks

are the top 30,000 customers who are likely to leave (or 'churn') annually. That is still only 30% of the 100,000 churners.

Of course, we can't assume that the model is completely accurate, so let's assume that it's only right 2 out of 3 times. That means that we've now correctly identified 20,000 customers who will churn annually. Remember however that we can only hope to persuade 50% of them to remain customers so that's 10,000 customers we have retained and 90,000 customers who cancel their contracts. These cancellations cost the company £5.04 million to replace and incur additional retention costs of £350K to persuade 10,000 customers to renew their contracts, bringing the total costs to £5.39 million. This represents a relatively modest cost reduction of £210K. Crucially though, in doing so they have managed to retain 10,000 customers with total annual revenues of £1.35 million.

A lot of clients see tasks like this as quite daunting. But the point here is that it's worth trying to figure out the impact that the application might make, so you can see from the outset if it's worth proceeding with the project. By doing this, you can often work out what the minimum threshold of success is. What if the model in our example could only hope to retain 5,000 or 3,000 customers? What if the retention costs were significantly higher, say £65? One thing is for sure. By the time the project is complete, someone in the organisation will go to the trouble of working out the costs and benefits and the results might not be so encouraging.

Other key outputs from this early phase of the project include creating a glossary of acronyms and jargon, identifying any constraints or restrictions such as security issues or legal requirements and making explicit the potential risks associated with the project being delayed or cancelled. It also makes sense that the stakeholders should estimate what resources they will need in terms of the required data, the hardware and software and the available personnel.

For my own part, at this stage of the engagement I'm usually leveraging my experience as well as my imagination to answer the simple question 'What will it take to make this application successful?' Initially, this involves trying to visualise how the data will need to look in order to generate accurate outcomes.

If we return to our earlier example of attempting to predict which customers approaching their contract end dates will renew them for another year, we can start to consider what kind of data we will need in order to generate our column of likelihood scores. Firstly, we need to think about the historical time window that we will be working within. How far back should we go to select a sample of data? Three years? Five years? This of course depends on the degree to which the customer base, the market conditions or the product offerings have changed in that time. It's certainly a question that I would ask the client to help answer.

Next we need to think about what proportion of people actually cancel their contracts annually, as ideally, we'd want the data to reflect this. Remember that we are trying to make a prediction at a particular point in the customer's contract i.e. three months

before it expires, so the sample of historical data should represent this.

Finally, we should focus on something called ‘the unit of analysis’. This simply refers to what a single row of data represents. For instance, a data record could represent an individual transaction, a website search, a session using an app or the demographic details of the customer. As the point of the application is to identify customers that are unlikely to renew their contracts, it makes sense that the unit of analysis should be individual customers rather than, for example, the multiple transactions *per* customer.

So, we’ll be working with a historical dataset where each row represents individual customers, three months before their contract ended. Let’s assume the data is taken from the last two years of the business and represents 1.2 million customers. We should also remember that in this example, the most crucially important field in the dataset is our analytical target ‘Contract_renewal’ which consists of two categories: ‘renewed contract’ and ‘did not renew contract’. Whatever technique we end up using, the job of the resultant model will be to accurately distinguish between these two outcomes so that we may apply it to current customers and take the appropriate action before the contract expires.

As for the remaining fields in the dataset, we may not at this stage know exactly what data we can access for the project, but we can begin to imagine what our ideal row of data might consist of as we attempt to predict the outcome in question. Figure 5 below shows a table of potential fields we might be able to use.

Customer Details	Customer History	Payment History
Gender	Tenure	Number of missed payments
Age group	Total service usage	Current payment method
Marital Status	Average service usage last month	Card expiry date
Postcode	Average service usage last 3 months	Bonus content last 3 months
License Type	Overall average service usage	Bonus content last 6 months
Device (Android, IOS)	Complaints	Contract renewal

Figure 5 - A sample of possible fields that might be utilised to predict contract renewal

As you can see, the potential fields are broken down into three groups: customer details, customer history and payment history. This is by no means an exhaustive list. It might well be that there are many other data sources and factors that we could use to predict our outcome, but it seems reasonable that the client might have data containing fields like these. Moreover, for all we know, demographic details like

gender might have no effect upon the target outcome field ‘Contract_renewal’ in which case this factor simply won’t be included in the predictive model. But given that it *might* help us predict the outcome accurately, it makes sense for us to request it as well as any other demographic factors that could be relevant such as customer age or marital status.

Variables such as ‘Average service usage last month’ and ‘Average service usage last 3 months’ may be useful in detecting whether the customer’s consumption of the service is starting to decline or if people who cancel their contracts use the service less than those who don’t. Payment history includes data related to the usage of bonus content. Fields like these could act as indicators of how invested the customer is with the service. To be clear, at this stage we’re only trying to imagine what data we might utilise as it helps to anticipate how the predictions could be generated. Until we move to the next phase of CRISP-DM we can’t usually confirm the availability of the information or make assumptions about its quality.

1.2.2 Making a plan

The business understanding phase requires that a project plan be drawn up. In my own experience of leading predictive analytics projects, a documented project plan, even if it’s not that detailed, is essential. It’s not uncommon for the original objectives of the project to seemingly evaporate over time. The basic task of getting the data into a decent enough shape to build a model can sometimes feel like hacking through a jungle and it’s all too easy to get lost. Whilst having an agreed plan that records the project’s rationale and objectives, as well as crucially, some kind of anticipated success criteria, may not guarantee a favourable outcome, it certainly helps to bring focus and agreement to the entire exercise from the very start.

I worked for three large technology companies over an 18-year period, leading experimental proof-of-concept projects. Due to the often ad-hoc and make-shift nature of the engagements, as well as the number of personnel from both organisations that were involved, there were plenty of situations where a single detailed, documented plan simply didn’t exist. Instead, sometimes the motivation seemed to be ‘let’s see what this will tell us’.

There’s nothing wrong with exploratory analysis for its own sake, as long as you’re prepared for the fact that the results might not be that interesting. If, on the other hand, the project is explicitly insight-focussed, then there should at least be an agreed list of the hypotheses to be tested. I’ve often been in situations where the stakeholders only had a hazy sense of how to evaluate the results of their own project. Even worse, sometimes they disagreed as to what the purpose of the exercise had been in the first place.

One of my most common experiences, however, is that showing the sponsors of the project the results seems to cause them to suddenly understand what their success criteria ought to have been. It is like being asked to paint someone’s living room when they’re reluctant to choose a shade of emulsion, only to be told when they see

the finished job that they hate the colour.

Examples of this include demonstrating how accurately a newly created predictive model performed, only to be informed by the client that they couldn't influence the outcome (meaning that the results weren't actionable); or evaluating a model that predicted the risk of a system error, only for it to become apparent that calculating the risk of an error wasn't as important as predicting the cost of the machine failing (so the wrong objective for the project had been chosen); finally, a horrified marketing manager learned that a successful predictive analytics project meant he would be expected to deliver more campaigns with smaller budgets (he hadn't been included in the initial project planning phase).

These situations were all avoidable, primarily by spending more time discussing and planning the project to ensure that everyone understood the potential ramifications of the application. Project outcomes like this are alarmingly common and that's something that I don't think is always well documented, or even mentioned, in the burgeoning literature of data science and AI.

CHAPTER 2

THE RAW MATERIAL

When we discuss potential predictive analytics applications with clients their first concern is often whether or not they have the necessary data resources to succeed. However, the success or failure of these engagements rarely comes down to issues with data. As we've already seen, factors such as failing to choose the right objectives, not stipulating success criteria, failing (or being unable to) act on the results and not evaluating the impact of the application are much more likely to jeopardise things.

Although the data understanding phase of CRISP-DM is designed to provide a documented assessment of the project's data sources, in actual fact most clients are able to informally check if any potential raw data are available before a project even begins. As part of that process, we would normally try to anticipate the kind of information that would be required for the application and advise them appropriately. This means that by the time we've reached the data understanding phase of the process, the project stakeholders already have a rough idea as to the nature and volume of the raw material we'll be dealing with.

2.1 Data understanding

As Figure 6 shows, the outputs from tasks associated with the data understanding phase are a series of reports detailing different aspects of the project data. Whilst it's not necessary to create physically separate reports for each task, it's important that all the stakeholders have a clear understanding as to the scope of the data and any issues with regard to quality or completeness.

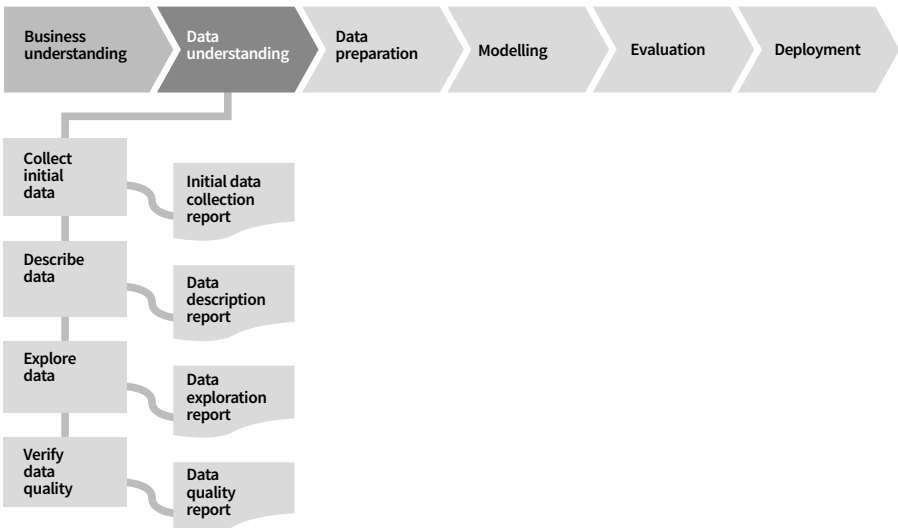


Figure 6 - Breakdown of the data understanding phase into tasks and sub-tasks

When I view client data for the first time it's generally a puzzle. Bear in mind that data can arrive in a huge variety of forms and volumes. A typical project's data haul might consist of a series of tables from a database, a collection of Excel workbooks, comma-separated or tab-separated text files or files saved in native formats such as SAS or SPSS. These files may originate in completely different departments and are often jumbled together in a series of archives.

A common theme is that the data have not been collected with any kind of analysis in mind. Although information from surveys, feedback forms, experiments or clinical trials is explicitly designed for analysis, in contrast, the data we tend to be working with have been collected for other purposes – such as campaign management, secure billing, stock control, asset registers, HR administration or CRM. These data often contain strange system fields, acronyms and short-hand terminology that mean absolutely nothing to those outside the business. Furthermore, the context in which the data are collected isn't recorded, so it's easy to misinterpret a piece of information or draw the wrong conclusions. In fact, it's not uncommon for the project sponsors themselves to be bewildered by this raw information. It may be the first time they've actually seen it. For these reasons the guidance of a subject matter expert is essential - someone on the inside who can explain what it all means.

The way in which the data are gathered can also add confusion. Ordinarily, a person in IT or a specific line-of-business is tasked with providing a data extract or series of extracts that are pertinent to the project objectives. Understandably, they usually ask what kind of extract the team want, how far back should it go and what fields should be included? The irony is that the project team might not know what these data extracts should include - they're guessing as well. A consequence of this is that we should expect some to-ing and fro-ing between the team and the data providers as the stakeholders start to gain a clearer picture of what relevant data the application can use.

One of the most useful things we can do when we first start to view potential data sources is to describe them. Creating a written summary of each file's purpose, the nature of the information, the data formats and the number of fields and records is a good start. We might also wish to point out any data that we have not been able to get hold of or that we are still waiting on. A typical risk to maintaining a project schedule is an unexpected delay in obtaining the relevant data. Having described the data at a high-level, we can start to investigate the fields and records themselves. The questions we need to address here include:

- Do any of the files contain multiple rows per customer (e.g. individual transactions)?
- Are there any fields that are irrelevant?
- Are there fields where it is not clear what information is being recorded?
- Which fields contain unusual or non-sensical values (e.g. minus numbers

for frequency counts)?

- How complete are the data in terms of missing values?
- Are there any highly skewed distributions that can affect summary measures like averages?
- Are there duplicate customer IDs?
- Are there any verbatim or open text fields?
- Are there fields that contain a large number of categories (e.g. many product codes)?

Figure 7 below shows a simple report that illustrates the kinds of issues that auditing the data can reveal.

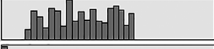



Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
Customer_ID		Nominal	1	889366	--	--	--	889366	889366
Age		Continuo...	5	111	42.626	14.481	-0.013	--	884133
Annual_Salary		Continuo...	-9.000	2600594.700	38591.354	19137.216	3.644	--	876263
Gender		Nominal	--	--	--	--	--	5	885009
Product_Code		Nominal	--	--	--	--	--	99	889366

Figure 7 - Report output showing summary measures for a data file with around 89,000 records

Looking at the summary measures for the field ‘Age’ we can see that the minimum value is 5 and the maximum is 111. We might well question whether or not these represent errors. In a similar manner, the ‘Annual_Salary’ field shows a minimum value of –9. This could be a deliberate entry to mark records where the information is unknown (commonly referred to as a ‘user missing value’).

The data file seems to contain at least one record where someone’s annual salary is around 2.6 million. This might not be an error, but obviously we should be aware of any extreme values that might affect our analysis. The bar chart associated with ‘Gender’ indicates five unique categories. Again, this is not uncommon when data comprised of character strings are inconsistently coded. For example, mixing upper and lower cases can lead to separate categories being created for a group recorded as ‘Male’, ‘MALE’ or ‘male’. Lastly, we should be aware that the field ‘Product_Code’ has 99 unique categories. It may be necessary to simplify this field if we want to make the analysis output easier to understand.

In a typical collection of project files many different fields are usually split across separate tables. For example, a customer contact file might contain information such as the names and billing addresses of the customer (or supporter, volunteer, patient etc.). Ideally, we would expect this file to contain only unique customer IDs and no

duplicates. The other files might record different aspects of customer behaviour such as their usage of a service (e.g. via a phone app), their transaction and billing history or their interaction with the business through a call centre. In these situations, we would need to address the fact that there are likely to be multiple rows per customer. Another common issue is that when we try to merge the details from the customer file with those in the transactional files we encounter mismatches. We might find lots of cases where the customer doesn't seem to have a transaction history (or vice-versa). Again, this should be documented because it may reveal an issue with how customer ID values are recorded as this can affect our ability to create a consolidated file against which to build a model.

The data understanding phase of the project is a good opportunity to sanity check the basic assumptions as set down in the project plan and to spot any relevant patterns that might affect the analysis.

- Are there enough examples of the key outcomes in order to build a good model (e.g. fraud)?
- Are there relationships that don't make sense?
- Is there an important trend or change in behaviours over time?
- Are there customers who shouldn't be included in the project?
- Are there any useful correlations between fields that might help predict the outcome in question?

Most students of statistics know that failing to thoroughly explore the data before beginning any serious modelling is regarded as something of a mortal sin. This is because exploring the relationships between key fields can help identify interesting patterns, especially when you are hoping to predict an outcome.

As an example, the table in Figure 8 shows that customer age group appears to have a bearing on whether or not they churn (i.e. cancel their contract). Only 43% of the customers aged '20 or younger' are still active compared to 60% of those aged 'Over 40'. This simple exploration indicates that age may be a good predictor of a person's likelihood to renew their contract.

Relationship Between Age and Customer Churn				
		Current Customer		
		Active	Churned	Total
		Row %	Row %	Frequency
Customer Age Group	20 or younger	43%	57%	8105
	21 - 30	53%	47%	10203
	31 - 40	61%	39%	6889
	Over 40	60%	40%	6557
	Total	54%	46%	31754

Figure 8 - Exploring the relationship between the customer age group and customer churn

Figure 9 below reminds us that CRISP-DM allows for a two-way relationship between the business understanding and data understanding phases of a project. This makes sense as these kinds of projects are often the first time that the data sources concerned have been brought together. As a result, the initial analysis may cause the stakeholders to go back and re-evaluate their thinking in the business understanding phase.

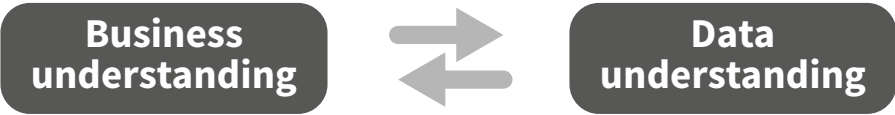


Figure 9 - The two-way relationship between business understanding and data understanding

Occasionally the output from this phase leads to the realisation that the organisation clearly doesn't have the necessary data to properly address the project objectives. In which case, the whole process is stopped. More commonly, the stakeholders have a better appreciation of the limitations of the available data and some sense of what needs to be addressed in order to prepare the files for model building.

The data understanding phase helps us to gauge how far away we are from our ideal data set. This would be a consolidated data source, with no errors or anomalous values, containing only fields which are potentially relevant to our objectives, comprised of a sufficiently large and unbiased sample of historical records from which we can draw reliable conclusions.

CHAPTER 3

SHAPING THE DATA

We've now planned our project in the business understanding phase and thoroughly audited our data sources in the data understanding phase, but we are still not ready to start deploying algorithms or generating models. There is some way to go before that can happen. The one thing that all analysts working in predictive analytics agree on is that the inevitable process of knocking the data into shape so that an effective model can be built is usually the one of most time consuming stages of the project.

3.1 Data preparation

The next stage of the CRISP DM process is the data preparation phase. The output of this phase includes a data cleaning report, the potential creation of new fields and records and, eventually, a fully documented, cleaned and consolidated data file. Many of these tasks can be quite resource intensive - so much so that it's estimated that the data preparation phase generally takes anywhere from 50% to 70% of a project's time and effort.

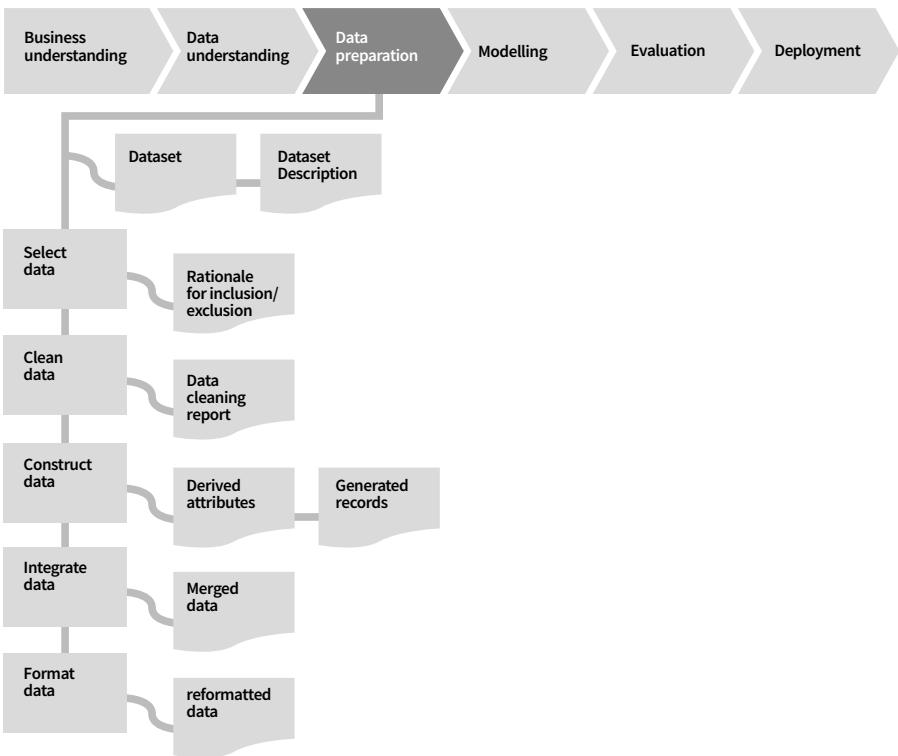


Figure 10 - Breakdown of the data preparation phase into tasks and sub-tasks

You won't hear much about data preparation from the people who sell predictive analytics technology. Most clients understand that some sort of cleaning and preparation will be required to make their data appropriate for analysis, even if they don't anticipate just how time consuming this process can be. However, it's difficult to demonstrate data preparation tasks in a way that senior executives are going to find interesting. Nevertheless, they should at least be aware that this is what data analysts spend a lot of their time doing.

To begin with, raw data is often messy and one of the immediate challenges we face is data cleaning. This in itself is a pretty broad area, but here's a list of the sorts of tasks you can expect to encounter when attempting to clean the data:

- Identifying records with values that don't make sense, such as quantities or amounts with minus numbers
- Removing duplicate cases
- Dealing with fields where a large proportion of the data are missing by estimating or 'imputing' the missing values
- Fixing date fields so that the system correctly identifies them as being dates
- Reducing the number of categories in a field so that we don't have too many individual values that occur less than one percent of the time
- Dealing with extreme values that have an adverse effect on summary statistics like averages by setting them to a maximum or minimum value
- Renaming fields so that they are easier to understand

Near the beginning of a project cycle, I start imagining what the data needs to look like in order to make the application work. What would a single row of data consist of? In our customer churn example, this would be a record representing an individual customer about three months before their contract renewal date. This row of data might contain a range of demographic details as well some fields related to the customer's payment and usage history (as illustrated in Figure 5 earlier). But how do we create this single row from multiple data sources? The simple answer is that we merge the files. The more complicated answer is that we find a way to merge the files so that any information loss is minimised.

Losing information is almost inevitable because unfortunately a perfect match between data files rarely exists. There are always customers that don't seem to have any transactions, or service usage records that don't match any customer ID value. Consider the issue of multiple rows per customer that we encounter when looking at each person's usage of a service or their transaction history. Given that we want to get to the stage where we can have one row representing all the necessary information about that customer, how can we turn all these rows of data into a single record without losing any detail? In most cases, you simply can't. You need

to summarise each customer's usage history. In our example that means we might calculate the average number of hours they spend per day or per week consuming content on a news app or playing a game. But an average isn't enough to summarise all that detail, so we may need to include the total number of hours they consumed in the first three months as well as the last three months. From here the number of measures we could take starts to increase exponentially. For example:

- The day of the week they spend most time consuming the service
- The minimum, maximum and standard deviation of average daily consumption
- The difference between the average number of hours of consumption in the first six months and the last three months
- The median amount of data consumed per day

These new measures will now appear as new fields in our merged data file. In the machine learning and data science community this is known as 'feature engineering'. A data analyst can easily end up creating a significant number of additional fields that measure different aspects of behaviour. But how are they supposed to know whether any of these new fields are necessary to drive a successful predictive analytics application? In fact, we don't know. Not for certain anyway. Instead we're trying to preserve different aspects of the historical information because it might turn out that one of these measures, or a combination of them, are important components of a future model.

Returning to CRISP-DM, there is another feedback loop (as Figure 11 below shows), this time between the data preparation phase and the modelling phase.

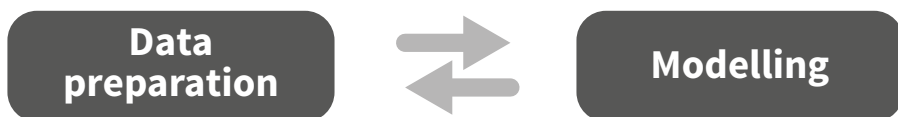


Figure 11 - The two-way relationship between data preparation and modelling

The model might show that a field needs to be cleaned up a bit more or that there is an interesting interaction between two factors. So, the analyst returns to their earlier data preparation work and attends to this before re-running the model to see the effects of the changes they've made. I once built a model to predict the likelihood that a patient would be re-admitted to hospital within 30 days of discharge. I realised that in order to predict the outcome accurately, I needed to create an index field that showed each patient's risk of readmission on the previous occasion they had been admitted.

This was a reasonably complex calculation that resulted in an index running from 0 to 10. The vast majority of patients had a value of 0 indicating they hadn't been admitted to hospital in the last 3 years. For the remaining patients that been admitted, a value of 1 indicated they were in the bottom 10% of risk and a value of 10 indicated they were in the top 10%. Not surprisingly, this new index field, which took account of their previous risk profile, became the most important indicator in predicting their current risk of readmission.

I've also found that sometimes calculating the ratio of one measure to another can be very useful. For instance, let's assume I notice that a model seems to make use of a field that records a person's income as well as another field that shows how many children they have. I might find by calculating a new field that roughly estimates their disposable income, taking into account their family size, that this new field correlates more strongly with the target outcome.

The data preparation phase of a project is where the team can start to think creatively about what indicators and measures could be added to the data to drive better results. Data Science students often spend a great deal of time learning how to 'tune' statistical algorithms and machine learning techniques to construct the most accurate, reliable models. But experience teaches us that transforming the data itself in order to optimise the quality of the information can mean that the modelling phase is much more straightforward.

To give you a sense of what other tasks the data preparation phase entails, here is a summary of the most typical procedures that need to be performed and a brief explanation of what each one does.

3.1.1 Merging and appending files

Merging data usually refers to adding additional columns of information. Separate files containing demographic data and transactional information can be merged to create a single, consolidated table using a common identifier field such as an ID number. This can be tricky as often there are mismatches between the files, and we need to bear in mind that there may be multiple rows of information for each customer in the transaction data.

Appending data refers to the process of adding together the rows from different files to create a consolidated data set. This means that (usually) both datasets contain the same fields, but the rows themselves may refer to different time points such as quarterly data, or different locations such as separate stores.

3.1.2 Aggregating data

Aggregation is the process of creating a summary dataset. This is most useful when we wish to combine multiple rows per customer to create a single summarised row. Earlier we discussed various ways in which we could summarise a field that recorded a customer's historical usage of a service such as a mobile game, a streaming media platform or a news app.

One of the ways we can do this quickly is by aggregating the data so that we end up with a single row for each customer comprised of summary measures such as the mean, minimum, maximum and total hours of consumption. This aggregated file, containing single rows for each customer, could then be merged back to the main customer contact file that contains all the useful demographic information about the customers. Something to bear in mind though is that aggregation doesn't work well with fields that can't be summarised using measures such as averages. For example, one can't calculate a meaningful average when the field shows the different genres of movies that people watched, or the devices that they used. For those kinds of situations, we tend to transpose the data.

3.1.3 Transposing data

In general terms, transposition of data refers to the process of turning rows into columns and vice versa. In this case however, we're using it to address a specific challenge. Imagine we have a field that shows all the different devices a customer uses to consume content on a streaming platform. The field might say they watched a series or listened to music on their phone in the morning, their laptop at lunchtime and their smart TV in the evening. Other customers might only use one device to consume the same content. This variation may turn out to be an important factor in their likelihood to renew their contract or subscription.

The problem is that each time a customer uses a different platform they create another row of information in the file that records service usage, but we only want to have one row per customer. A simple way to resolve this is to transpose the data so that each device is represented by its own unique field. A value of 1 in the 'Phone' field would mean that the customer has used that kind of device whereas value of 0 in the 'Laptop' field means that the customer hasn't used that device. By creating these 'flag' fields, we can represent the usage of multiple devices in a single row.

3.1.4 Creating new fields

Whilst the process of aggregation and transposition can result in creating lots of new fields, we can also run procedures with the explicit aim of deriving new fields. Examples might include calculating the number of elapsed days between two date columns; creating a field such as 'age group' based on the customers' ages; generating a series of columns that show percentage spend rather than actual spend; calculating a field containing an overall mean score based on a group of numeric fields; or creating a new column that indicates whether or not a customer is regarded as high value.

Data preparation devours project time like no other activity in the CRISP-DM cycle, but that's because it's so important. Despite the apparent sophistication of so many statistical and machine learning techniques, in reality these methods work best when the input data has been thoroughly prepared before they are 'spoon-

fed' to the model-building engine. This can make a huge difference to the overall utility of the resulting application.

CHAPTER 4

THE ALGORITHM MENAGERIE

We now turn our attention to modelling techniques. However, before we directly address the modelling phase of CRISP-DM, let's take some time to consider the range of analytical approaches we can use to build a predictive analytics application.

Remember that we are working towards a point where we're hoping to generate new and useful information. Depending on the application, these data can take many forms - predicted outcomes, estimated values, cluster groupings, anomaly scores or forecasted trends to name but a few.

The word 'algorithm' is often loosely employed in the media to denote a set of mathematical rules that make recommendations and drive decisions in machine learning and AI applications. Traditionally however, in disciplines like applied statistics, an algorithm refers to a procedure or technique that interrogates a file to generate or uncover a model of the relationships in the data.

Using an algorithm like linear regression, for example, results in a model that is actually an equation we can use to estimate a value. Other kinds of algorithms might produce a model that comprises a set of logical rules that also are used to estimate (or predict) a value.

The point is that within the literature of analytics, 'algorithms' refer to techniques that create models and it is these models that generate the new columns of data. Most of the model types that generate these new data fall into three broad groups, which can be summarised as prediction, segmentation and association. Let's take a moment to examine each of these families of modelling techniques in order to better understand how they are applied to solve real world problems.

4.1 Predictive models

Predictive algorithms usually need data sets in which the data rows record different situations resulting in a specific outcome such as someone cancelling a contract, or spending money on a product, or contracting an illness or expressing satisfaction with a service. Using these historical data where the outcome of interest has already happened, a predictive algorithm can be used to train a model to predict a future result.

It does this by combining the data from input fields to create a pattern or equation that either estimates the outcome directly or calculates the likelihood that a data row will belong to a particular category in the target field. These techniques can therefore be used when the outcome (or target) field takes the form of a number, such as the value of an insurance claim, a crop yield or the total amount spent in a transaction. But many methods can also be used when the target field is comprised of category outcomes, such as an account status or the response to an offer².

² Numerical fields measuring things like amounts are usually referred to as *continuous* whereas those with category outcomes are referred to simply as *categorical*

Whether the predictions are in the form of estimates, probabilities or classifications, the project team must carefully evaluate them within the context of the success criteria they specified in the business understanding phase. There are well over a hundred different predictive methods available to most analysts but as we shall see, they all tend to belong to one of three primary types.

4.2 Segmentation models

Unlike predictive techniques, segmentation algorithms don't rely on target or output fields. Instead, they're designed to identify segments, or clusters of records that are similar to one another. Because these techniques are not typically used to train a model against a key outcome, they only require input fields.

In predictive modelling of course, the target field represents the answer that the technique is trying to 'learn'. For this reason, predictive modelling is often referred to as supervised learning so, conversely, techniques like segmentation are referred to as unsupervised learning.

Segmentation (or clustering) models are useful when attempting to characterise a customer database in terms of demographics and shopping behaviour in order to drive deeper customer insights and provide more compelling products or services. Other applications include using data from smart meters to establish the various ways in which different kinds of households consume electricity.

I've worked on applications where clustering was employed to group similar retail stores together in terms of their sales patterns in order to more effectively resource them and to uncover different modes of shopper behaviour. Such models are evaluated not in terms of their accuracy but rather in terms of their ability to capture interesting groupings in the data that might otherwise have gone undetected. Rather than the resulting output being a column of data containing predicted outcomes, it's a cluster membership field indicating the segment (or cluster group) to which each row was assigned.

One of the biggest problems with these kinds of models is that often there is no correct result. To begin with, the analyst must figure out what makes one cluster grouping different from another. Sometimes the differences are quite trivial, or the segments vary wildly in size with 90% of the records being assigned to one cluster group and less than 1% to another.

Even the number of clusters uncovered may vary from one analysis to another, especially because, in most cases, the analyst has to specify in advance how many cluster groups the procedure should produce. Most analysts try several different iterations and compare the resulting segments until they select a solution that they regard as interesting or useful. Ultimately, this can be a completely subjective judgement and so it's open to criticism by potentially sceptical stakeholders.

A related procedure to cluster analysis is anomaly detection. Here the focus is not on finding groups in the data that appear to naturally coalesce, but instead on

identifying cases that are hard to assign to a cluster. In repeated cluster analyses, these cases may seem to stubbornly ‘want’ to form their own clusters. It may be that they are doing so simply because they are anomalous.

Some anomaly detection techniques exploit this tendency by measuring how typical a case is with respect its peer group (or cluster). The procedure can then generate an anomaly index field indicating the degree to which each case is unusual. Anomaly detection is therefore particularly useful for detecting clandestine or fraudulent behaviour. It does however suffer from the same issues that plague cluster analysis, in that there is more than a degree of subjectivity involved. Certainly, in fraud analytics one cannot assume that something is suspicious just because it is anomalous.

4.3 Association models

Association models comprise simple sets of rules that aim to uncover relationships between different categories in the data. These rules might be as simple as ‘IF bread THEN butter’. Association algorithms differ from the standard predictive approaches in that the fields can act as both inputs and targets.

With this kind of approach, we are more interested in the interrelationships between the variables and categories rather than the outcomes recorded in a single field. Association models are useful when we wish to understand the co-occurrence of events. The output from these models commonly take the form of new fields, with each showing a category suggestion and an accompanying ‘confidence score’. In the retail sector, for example, these techniques are often used to perform basket analysis or affinity analysis.

By using association algorithms to uncover sets of products that are purchased together, retailers can design new bundle offers or even alter their store layout to enhance convenience and encourage shoppers to make impulse purchases. A similar approach is used by companies like Amazon who make product suggestions based on past purchase history. Beyond the retail sector, association algorithms are used to identify component parts that are likely to fail together, symptoms that tend to co-occur based upon medical conditions or patterns of behaviour in events linked to fraud and crime.

An extension of this capability are sequence detection algorithms which aim to find useful sequential rules in time-structured data. Such algorithms are often deployed against asset maintenance data to determine patterns of failure. Here, the order of the fault occurrence is key. For instance, sequence rules might show:

IF oil pressure warning FOLLOWED BY temperature light THEN pump failure.

As opposed to:

IF temperature light FOLLOWED BY oil pressure warning THEN replace oil

In the first rule the eventual discovery of an oil pump failure is indicated by an initial drop in the oil pressure and a subsequent rise in the machine temperature due to increased friction. In the second rule the diagnosis that the oil needs replacing is

associated with an increase in the temperature due to a of loss viscosity (associated with old oil) and a subsequent drop in pressure.

4.4 Other model types

There are of course several kinds of algorithms that don't fit neatly into any of these three main groups. They include time series techniques which attempt to model trending data in order to forecast into the future. Such models are used for everything from forecasting website traffic and product sales to demand for electricity and air travel passenger volumes.

Also of note are text mining algorithms that categorise unstructured data such as open-ended responses from surveys, hotel reviews, error messages and emails. These categories can then be converted into fields that denote the different topics under discussion.

In fact, text analytics is a very powerful way to create data that can be employed in predictive models and association analysis. I worked on a project that mined the text both from system errors and from the subsequent engineer reports that were created when the problem was fixed. By using an association algorithm, we were often able to predict what was required to resolve the problem, including which parts were likely to need replacement, before the engineer was even despatched.

4.5 The two cultures

So far we've discussed the primary types of approaches to modelling in terms of the kind of problems they address. But the techniques that populate these approaches have quite different pedigrees. Whilst the term 'machine learning' is commonly used, especially in the media, we've already established that predictive analytics is actually a more accurate description for these kinds of capabilities because it encompasses a number of analytical approaches, including machine learning and classical statistics, with a focus on generating new data to aid decision making.

So why has machine learning caught the media's imagination? To my mind, there are two main reasons. Firstly, computer scientists working with machine learning algorithms have made a number of high-profile breakthroughs over the last few decades, particularly in the area of image recognition systems. Secondly, it sounds cool.

In the 1992 movie *Terminator 2: Judgement Day*, the time-travelling, android assassin played by Arnold Schwarzenegger utters the chilling line 'My CPU is a neural-net processor; a learning computer. The more contact I have with humans, the more I learn'. It probably wouldn't have sounded as impressive if he'd mentioned logistic regression. And yet, techniques like logistic regression are often lumped in with machine learning despite the fact that its usage can be traced back to the early 1940s when its application was calculated more or less manually.

That wasn't unusual for the time. Despite the fact that pretty much everyone working in applied statistics now uses computer software to calculate their results, statistics

actually pre-dates modern computing by several decades. That's because the discipline of statistics is characterised by its reliance on probability distributions which are pre-calculated and ready to be applied to a host of applications and problems. In other words, you don't need a computer in order to do statistics.

Machine learning however, as the name implies, does require computing power in order to uncover useful patterns among apparent randomness in large datasets. This is what the statistician and political forecaster Nate Silver refers to as 'the signal and the noise' (Silver, 2013). Machine learning is traditionally associated with computer science, and not surprisingly, its evolution as an analytical approach closely mirrors the progress of computing technology from early 1950s until the present day.

Many of us who worked in the analytics software sector in early 1990s have clear memories of the level of hostility that some members of the statistical community exhibited towards machine learning. The world of statistics is based on a very different philosophy and cultural outlook than the world of machine learning.

In particular, statisticians loathed the opaque models that machine learning methods such as neural networks produced. For instance, whilst Warren Sarle's 1994 paper, *Neural Networks and Statistical Models* (Sarle, 1994), provides an excellent primer in these algorithms, he does little to hide his personal disdain for them. In the conclusion he writes *'The marketing hype claims that neural networks can be used with no experience and automatically learn whatever is required; this, of course, is nonsense'*. Although it's not clear what 'marketing hype' he was referring to, he nevertheless confidently concludes *'It is therefore unlikely that applied statistics will be reduced to an automatic process or 'expert system' in the foreseeable future. It is even more unlikely that artificial neural networks will ever supersede statistical methodology'*.

Given that some of our most far-reaching technological advances such as using image recognition to detect skin cancer, real-time spoken language translation and the development self-driving cars are all underpinned by the algorithmic descendants of neural networks, it's ironic that Sarle (1994) was making a prediction about something that is, in essence, predictive...and at the same time getting it so embarrassingly wrong.

Many statisticians in the 1990s still viewed machine learning as 'the new kid on the block' and, at the same time, its promoters seemed wilfully ignorant of hard-won statistical conventions. Never mind the fact that many of the problems machine learning claimed to solve had already been thoroughly addressed using statistical methods. To machine learning enthusiasts, the statistical community, with their emphasis on populations and samples, their arsenal of obscure distributions and their quaint insistence on correctly framing a problem within its theoretical context, simply weren't relevant to the kinds of challenges these newer approaches aimed to address.

The two schools of thought even seemed to have different languages to describe the

same things. Figure 12 illustrates this in more detail. By the way, up until now I have deliberately chosen to describe a column of data denoting factors like a person's gender as a 'field'. You should be aware that in statistics, the equivalent term is 'variable'.

Statistical term	Machine learning term
Variable	Feature
Parameter	Weight
Fitting	Learning
Dummy coding	One hot encoding
Regression/classification	Supervised learning
Cluster analysis	Unsupervised learning
Dependent	Target
Independent	Predictor

Figure 12 - Statistical terms and their machine learning equivalents

The foundations of classical statistics are rooted in an attempt to quantify the likelihood of an occurrence or a general truth given limited amounts of data. These limited data take the form of samples. Indeed, statisticians might argue that *all* datasets are samples, in that they are almost always a snapshot of something where there are many other possible observations.

In this sense, samples are used to make *inferences* about wider populations. In exactly the same way, scientists might take a small sample of sea water to make a judgement about what is happening in the wider ocean. But given the fact that each time they take a sample, the results might be slightly different, how can they estimate things like the median temperature or average salinity of an entire ocean from a limited number of observations?

Suffice to say that decades of work have resulted in a stream of techniques and approaches that aim to solve exactly these kinds of problems. At the heart of all these techniques is the use of probability. In particular, statistics as a discipline is characterised by the use of probability distributions to estimate how often something might occur or how reliable the observation of a particular relationship might be.

Ultimately, both of these situations require knowledge of the population. But that's like expecting an oceanographer to have complete knowledge of the entire ocean. It's not possible. So instead, probability helps us to infer how likely it is that the observations in our (hopefully unbiased) sample are simply reflecting the realities in the wider population (or ocean). Moreover, because we can never truly know what's happening in the population, the probability can be never 100% or 0%. It is always somewhere in between.

If you find the idea of probability complex and mysterious, then you are in good

company. In his excellent book, *The Art of Statistics*, the renowned British statistician, Sir David Spiegelhalter candidly remarks, *‘I am often asked why people tend to find probability a difficult and unintuitive idea, and I reply that, after forty years researching and teaching in this area, I have finally concluded that probability really is a difficult and unintuitive idea. I have sympathy for anyone who finds probability difficult.’* (Spiegelhalter, 2019)

Statisticians use probability calculations to account for the variation observed in samples drawn from populations. In any empirical science, being able to effectively ‘account for variation’ means that one should be able to make predictions. By studying the factors that affect the heights of 100 marigold plants, you might discover that sunlight, ambient temperature and water are key independent variables that are related to the variation in this dependent variable outcome. Statistical procedures can then be used to develop predictive models that represent the relationships between these factors. These procedures are underpinned by a wider theoretical context that attempts to take into consideration the shape of the distribution of marigold heights and the likelihood that the differences we are observing are also present in the theoretical population of all possible marigolds rather than simply being artefacts of our sample.

This is the kind of context that statistical prediction operates in. The machine learning approach on the other hand, could be more plainly characterised as using a computer algorithm to find a pattern in the marigold data so that you can generate accurate predictions.

4.6 Statistical techniques

The majority of the most popular statistical techniques used in predictive analytics applications are members of the regression family. Of these, arguably the oldest and most widely used predictive algorithm in data analysis is linear regression.

The term ‘regression’ is shorthand for ‘regression to the mean’. This relates to the idea that individual observations may vary greatly from one another, but that their overall mean value is close to the same number when we draw repeated samples. In a similar way, the scatterplot in Figure 13 shows the relationship between horsepower and engine size for a sample of cars.

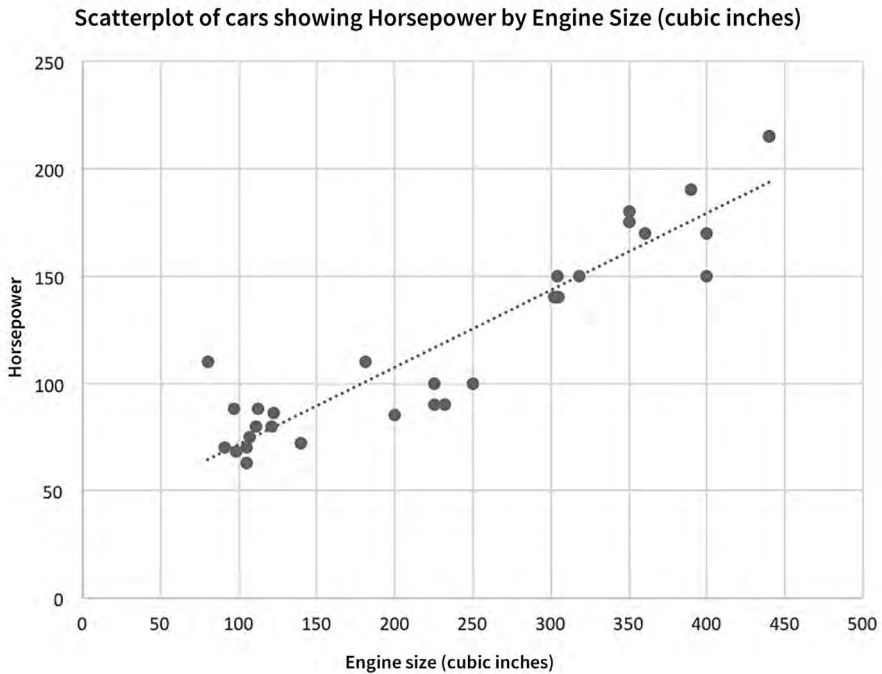


Figure 13 - Scatterplot showing relationship between horsepower and engine size for a sample of cars

The chart shows that cars with larger engines tend to have more horsepower and vice versa. The dotted line running through the data points is effectively the average of this relationship, and in this sense, the data points cluster about the mean. The line itself is automatically fitted in such a way so that the distance between the points themselves and the straight line is as small as possible. For this reason, it's referred to as the line of best fit.

Furthermore, because the relationship shown in the scatter plot seems to be reasonably well represented by a straight line, we might state that, based on these data at least, the relationship appears to be linear. We could then use this regression line to predict, say, the horsepower for a car with an engine size of 250 cubic inches, just by looking up on the chart where the value of 250 on the horizontal axis intersects with the diagonal fit line and then read off the corresponding value on the vertical axis (about 125 hp).

An alternative approach is simply to use the equation for a straight line as the basis of the prediction. This simple equation is normally expressed as $y = mx + c$. In this case y represents the car's estimated horsepower and x would be the hypothetical

engine size. The term **m** refers to the gradient of the line and the value **c** is where the line cuts the vertical (or y) axis. As it turns out, for our chart the values of **m** and **c** are 0.359 and 35.5 respectively, so the equation would be:

$$\text{Horsepower} = (\text{Engine Size} \times 0.359) + 35.75$$

Which means that the estimated horsepower for a car with an engine size of 250 cubic inches would be 125.45. This simple equation is the basis of linear regression, and as such we can regard it as the foundation of predictive modelling.

However, like most statistical procedures, linear regression makes a number of assumptions about the kind of circumstances in which it is applied. It doesn't work well, for example, when trying to estimate a category outcome such as whether or not a baby will be born with a healthy weight or if someone is sufficiently creditworthy to be lent a sum of money. For those types of situations, statisticians often reach for logistic regression. Logistic regression attempts to estimate the probability that a particular category outcome will occur based on the values of one or more independent variables.

There are other situations, such as data that counts the number of times an event occurs, like visits to the dentist or insurance claims made in a year, where more specialised regression procedures known as generalized linear models may need to be applied. Then there are non-linear regression techniques that might be applied in circumstances where, for example, the analyst is modelling some kind of growth curve (perhaps for marigold plants?).

Whichever method is employed, statistically-based prediction is characterised by the use of probability tests to detect important factors before expressing the resultant models in the form of equations. The fact that many of these techniques were developed long before electronic computing became widely available, and that they continue to yield surprisingly accurate results, is testament to the collective ingenuity of the community that developed them and to the pragmatic richness of statistics as a discipline.

4.7 Rule induction / decision trees

Rather than expressing predictive models as formulae, an alternative class of techniques uses logical rules to find distinct groups in the data that exhibit different outcomes. Rule induction is a technique that creates 'if-then-else' type rules from an array of input variables showing the most likely outcome. This approach is also the basis for the association models that we encountered earlier in this chapter.

One group of algorithms that make particularly good use of logical rules are decision trees. Decision trees build highly visual models revealing how the data can be hierarchically split into groups and sub-groups. In reality, these models usually appear as *inverted* trees, so in fact they look more similar to a root system where the rules tunnel their way through the data. The branching rules are automatically created by the algorithm to find segments that maximally discriminate between the

levels of a dependent target variable. In other words, they define groups in the data that are as different as possible from one another in terms of the target outcome.

Some decision tree algorithms can work with continuous target fields, but decision trees are more commonly deployed against targets containing two or more categorical outcomes. Typical target variables might record responses to marketing campaigns, customer churn, satisfaction levels, inspection outcomes or risk categories.

Decision trees are popular techniques because they produce transparent and intuitive models that are relatively easy to control and understand. Figure 14 below shows a decision tree built using the statistically-based CHAID algorithm.

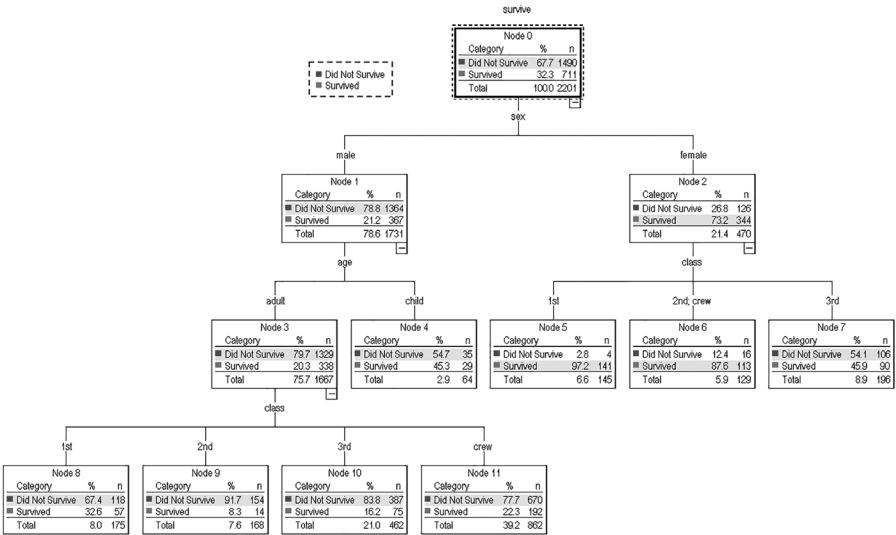


Figure 14 - Example decision tree using the CHAID algorithm to predict survival on the RMS Titanic

CHAID stands for 'Chi-Squared Automatic Interaction Detection' and it was designed in 1980 by Gordon V. Kass (Kass, 1980). It is an especially popular technique because it's based on one of the most widely used tests of statistical significance - Pearson's Chi-Square test.

The CHAID decision tree shown above aims to identify the most important factors that influenced survival on the RMS Titanic in 1912. Using the results of a Chi-Square test, it chooses gender as the biggest 'predictor' of survival. We can see why it might choose this variable by looking at the values displayed in the box at the top of the tree. This root node indicates that of the 2,201 passengers in the dataset, only 32.3% (711 passengers) survived the disaster. The decision tree algorithm then chooses the variable 'sex' to split the data into two groups. We can see that of the 1,731 male passengers in Node 1 only 21.2% survived, but of the 470 female passengers in Node 2, 73.2% survived.

Having established these two groups, the tree is further split into subgroups. Initially, the male branch of the tree is split by the variable 'age', whilst the female branch of the tree is split by the variable indicating the passengers' class of travel. According to the algorithm, the reason the female branch ignores the age of the passengers is because the survival chances of female passengers were not affected by whether they were adults or children. This is how the algorithm detects interactions between the variables in the data and finds exceptions to more general rules. The tree continues to split into subgroups until it either runs out of data or encounters an internal procedural rule that means it can go no further.

Not all decision tree algorithms use statistical tests to choose variables for inclusion in the model, but each technique results in a tree where every data record ultimately ends up in a final subgroup (or terminal node), and we can use these final subgroups to predict the outcome of interest. This means decision trees can work well as profiling tools and also as predictive models.

The tree itself is of course a visual representation of the rules from which it's comprised. Figure 15 shows the two rules that describe what the groups with the highest and lowest chance of survival might look like. Note that the probability of survival is simply the proportion of cases in each of the two subgroups that survived, so that for example, 97.2% is expressed as 0.972.

```
/Rule1  
IF sex = 'female' AND class = '1st' THEN  
NODE = 5  
PREDICTED_OUTCOME = 'Survived'  
SURVIVAL_PROBABILITY = 0.972.  
  
/Rule 2  
IF sex = 'male' AND age = 'adult' AND class = '2nd' THEN  
NODE = 9  
PREDICTED_OUTCOME = 'Did Not Survive'  
SURVIVAL_PROBABILITY = 0.083.
```

Figure 15 - Decision tree rules showing predictions for the two groups with the highest and lowest chance of survival respectively on the RMS Titanic

Decision trees based on rule induction methods have given rise to several modifications whereby not just one tree is created in a single model, but sometimes hundreds. For example, random forest algorithms, developed by statisticians such as Leo Breiman (Breiman, 2001), work by generating many individual decision trees that have been trained against different random subsets of the data. Each subset may consist of a random selection of the data records as well as the input variables, so that no two data subsets are identical. The resulting 'forest' of trees is then

aggregated together to create a single combined prediction.

The idea of a random forest is to prevent decision tree models from overfitting. This is a problem that occurs when the resultant decision tree too closely mirrors the idiosyncrasies of the sample dataset and so does not work well when required to make predictions against new data. Techniques like random forests make the sample dataset appear as if it is in fact many different samples so that no one single decision tree can dominate the final model.

4.8 Machine learning

As we've already discussed, the term 'machine learning' is commonly used to refer to a huge variety of multivariate techniques, many of which are actually purely statistical in nature. In this section however, we will look at the kinds of algorithms that fall within the more narrow, technical view of machine learning.

One of the things that makes machine learning distinct from classical statistics as an approach, is its lack of reliance on a host of probability distributions that pre-calculate the likelihood of different kinds of outcomes or events. Instead, machine learning algorithms assume very little about the how the data values are distributed, making use of computer programs that can automatically discover patterns within the dataset that yield accurate predictions. These learning algorithms are trained against sample datasets and the resulting models are then checked for accuracy against a test dataset.

Machine learning capabilities have gathered pace as the availability of computing power has accelerated. In 1951, Marvin Minsky and Dean Edmonds built the Stochastic Neural Analog Reinforcement Calculator (or SNARC) - a physical machine that acted as neural network and was, according to Popular Mechanics, '*capable of machine learning at a time when most computers still ran on punchcards*' (Wenz, 2016).

In the 1960s, neural network programs comprising multiple layers were developed, further enhancing the capabilities and range of applications to which these algorithms could be applied. By 1967 the Nearest Neighbour algorithm, used to map optimal routes in navigation systems, was created. In 1989 the first commercially available software using genetic algorithms was released. By the mid-1990s predictive algorithms such as Support Vector Machines began to emerge. Since then the number of machine learning applications has grown exponentially, especially in those areas of AI that mimic human abilities, such as speech and image recognition.

It's reasonable to wonder where all of these advances leave traditional statistics as an alternative approach. Certainly, key figures in the development of non-traditional analytical techniques such as Leo Breiman were not afraid to vent their criticism of the statistical community's reluctance to engage with the march of machine learning. In 2001 he wrote '*The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory,*

questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets.' (Breiman, 2001)

A few years earlier Jerome Friedman, a pioneer of rule-induction methods and Brieman's one-time collaborator in the development of the landmark CART (Classification and Regression Trees) algorithm, wondered if the two approaches might not have missed an opportunity to find common ground, writing '*Had we incorporated computing methodology from its inception as a fundamental statistical tool (as opposed to simply a convenient way to apply our existing tools) many of the other data related fields would not have needed to exist. They would have been part of our field.'* (Friedman, 1997)

Critics of machine learning argue that these kinds of algorithms produce models that are too opaque to directly interpret. You can check the accuracy and view the results, but you can't always make sense of *how* or *why* the model generated a particular outcome. Moreover, they argue that this lack of transparency can have potentially serious ethical and safety ramifications. Furthermore, statistical methods remain the de facto approach to help us separate what might be random effects in data from non-random. It's the analytical foundation of how we find things out.

For most analysts working in predictive analytics, machine learning algorithms are just another set of tools that they can call upon as required. There's no reason to assume that they will produce superior results when compared to more traditional statistical or rule-based methods. It therefore makes sense to keep an open mind as to which technique will address a project's objectives most effectively.

Figure 16 summarises the three main approaches that we have discussed thus far, as well as some examples of the algorithms that one may encounter in each class of technique.

Approach	Overview
Statistical	<p>Derived from traditional statistics and often based on some form of regression. They include:</p> <ul style="list-style-type: none"> • Linear regression, logistic regression • Generalized linear models (GLM) • Discriminant analysis • Factor analysis/structured equation modelling • Survival analysis
Rule induction / decision trees	<p>Rule induction algorithms use criteria from statistics and machine learning to derive sets of rules and trees. They include:</p> <ul style="list-style-type: none"> • Classification and regression trees (CART) • CHAID • C5 • Gradient boosted trees • Random forests <p>Association and sequence algorithms include:</p> <ul style="list-style-type: none"> • CARMA • Apriori • SPADE
Core machine learning	<p>Derived from computer science research. They include:</p> <ul style="list-style-type: none"> • Neural networks • Support vector machines (SVM) • Deep learning • Genetic algorithms

Figure 16 - The three main families of analytical approaches with example algorithms

CHAPTER 5

BUILDING A PREDICTIVE MODEL

Not surprisingly, the logical first step in the CRISP-DM modelling phase is to choose a model-building technique. The initial criterion for making this choice relates to the project's business objectives. For instance, is the purpose of the project to segment customers, provide product recommendations, identify anomalies, create forecasts or predict an outcome? The answer to this question will guide us to the most appropriate model-building methods. If, for example, the project requires us to predict an outcome, we need to know whether this outcome is a continuous value or a category. As we answer these questions, at each step, we are narrowing down the pool of techniques that we can use.

We may then need to think about the assumptions that each method makes about our prepared data. Statistical procedures may only be appropriate for data that meet certain distributional assumptions. Other procedures may perform poorly with missing data or only work with continuous predictor fields.

Understanding the various limitations and characteristics of different algorithms is part and parcel of the job that data analysts do. For myself, once I've established whether the purpose of the project is to, for example, predict an outcome or cluster data, I try to choose a procedure that is reasonably flexible and produces results that are relatively clear. When working on a project where the objective is to predict a categorical outcome, I might for example, choose to work with an interactive CHAID decision tree for precisely these reasons.

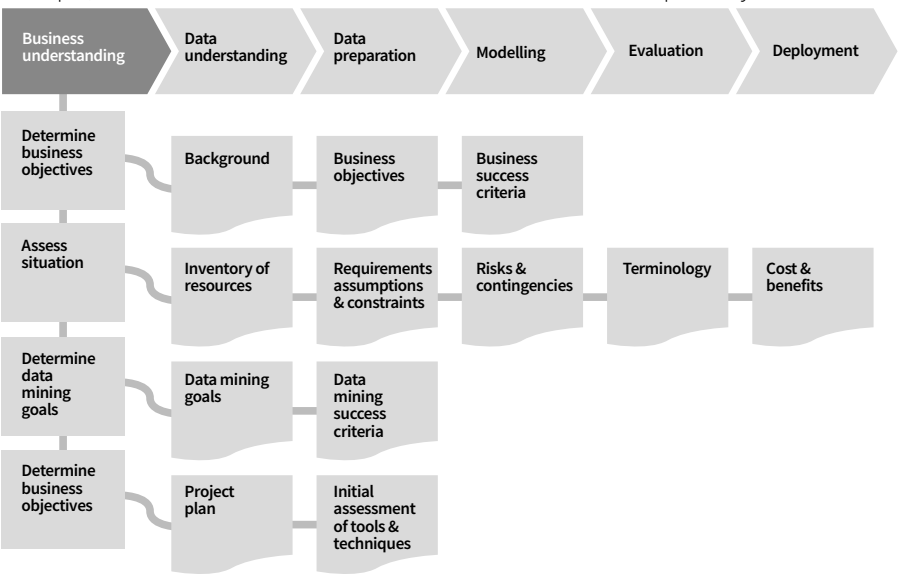


Figure 17 - Breakdown of the modelling phase into tasks and sub-tasks

As Figure 17 shows, the next thing we need to bear in mind is how we intend to test whatever model we finally develop. This, in CRISP-DM terms, is referred to as generating a test design.

One of the simplest ways to do this is to define a random subset of the data for testing purposes. This could take the form of creating a random field in the dataset that indicates whether a case is part of the test group or the model train group. An alternative approach is to create a physically separate dataset that has been randomly drawn from the original data file, purely for the purpose of testing the model.

The conventional approach is to train the predictive model against a healthy slice of the randomly selected data, say 70%, and then test the accuracy of its predictions against the remaining 30%. These days however, analysts are just as likely to use a 50/50 split. In either case, we are assuming that we have enough data to build and test the model to begin with.

That said, I have encountered situations where the project had less than 50 rows of data to build a model against. In these situations analysts often have to experiment with more exotic workarounds such as data simulation or testing methods such as 'leave-one-out cross validation' where the model is repeatedly built against all but one of the records in the dataset, and then used to predict the outcome of each withheld record in turn.

Let's begin with an illustrated example. Figure 18 shows a screen grab from a data file containing around 7,000 customers of a telecommunications and media company. Each row represents a different customer. The file itself has been created as a result of cleaning and merging various data sources during the data preparation phase of a project that aims to predict customer churn. We can start the process of predicting this outcome by identifying the 'Churn' field, shown at the end of the file, as our target field (or dependent variable). In actual fact the file contains a further 24 potential predictor fields (or independent variables), not all of which are shown.

customerID	gender	Tenure	Insurance	Box_Office_Movies	Auto_Renew	Monthly_Rate	Total_Bill	Churn
2227	Male	58	No intern...	No internet service	One year	20	1109	No
2228	Female	70	No	Yes	Two year	104	7349	No
2229	Male	4	No	No	Month-to-month	71	294	No
2230	Male	45	No intern...	No internet service	Two year	20	929	No
2231	Male	10	No	No	Month-to-month	70	740	Yes
2232	Male	36	No intern...	No internet service	Two year	19	755	No
2233	Male	54	No	No	Month-to-month	70	3883	No
2234	Male	23	No	No	Two year	60	1414	No
2235	Female	41	Yes	No	One year	78	3211	No
2236	Male	5	No	No	Month-to-month	71	372	No
2237	Male	27	Yes	Yes	One year	46	1246	No
2238	Female	1	No	Yes	Month-to-month	96	96	No
2239	Female	67	No	No	Two year	36	2546	No
2240	Female	72	Yes	Yes	Two year	90	6449	No
2241	Female	56	No intern...	No internet service	Two year	25	1469	No
2242	Male	44	No intern...	No internet service	One year	25	1014	No
2243	Female	66	No	Yes	Month-to-month	101	6691	No
2244	Female	34	No	No	One year	64	2089	Yes
2245	Female	69	Yes	Yes	Two year	105	7241	No
2246	Female	1	Yes	Yes	Month-to-month	102	102	Yes
2247	Female	40	No intern...	No internet service	Two year	20	830	No
2248	Male	30	No	No	Month-to-month	54	1589	No
2249	Female	11	No	Yes	Month-to-month	71	829	No
2250	Male	11	Yes	Yes	Month-to-month	69	712	No
2251	Male	72	Yes	No	Two year	74	5361	No
2252	Male	72	Yes	No	Two year	93	6735	No

Figure 18 - Selection of fields from a prepared data file used to build a model predicting customer churn

We will use the interactive CHAID decision tree algorithm to build an initial model. This algorithm begins the model-building process by scanning all of the 24 predictor fields and using a Chi-Square statistical test to choose the variable that discriminates ‘best’ between customers who churn (i.e. cancel their contracts) and those who remain subscribers.

At this stage, CRISP-DM suggests that we choose the appropriate parameters in the algorithm to build the model. This refers to the settings that govern how the model is built. However, the CHAID algorithm already has some sensible default settings so we can simply run the procedure. Because we are using an *interactive* decision tree, we can see what the algorithm is doing at each stage of the process and even overrule it if we feel it’s necessary to do so. It’s worth noting that in this example, only a random 50% of the 7,000 rows of data has been made available for the model-building process and that the software used in the images is IBM SPSS Modeler v18.2. Figure 19 shows the first stage in the model-building process: the root node of the decision tree.

Churn

Node 0		
Category	%	n
■ No	73.513	1829
■ Yes	26.487	659
Total	100.000	2488

Figure 19 - The root node of the CHAID interactive decision tree

We can see that the root node contains two categories. The 'No' group indicates customers who have not churned and who remain subscribers. These account for 73.5% of the sample training data. The 'Yes' group of course represents those subscribers who have cancelled their contracts and churned. The column marked 'n' shows the actual frequency count for each group.

In total, the decision tree is using 2,488 records to build the model. In case you are wondering, I will explain later why this number is not closer to the expected 3,500 which would represent exactly 50% of the original 7,000 records. At this stage, the algorithm attempts to choose the most statistically significant predictor variable to split the data. In fact, as Figure 20 shows, we can view which variables it is considering entering into the model next.

Select Predictor				
Predictor	Nodes	Statistic	DF	Adj. Prob.
Auto_Renew	3	Chi-square=425.149	2	0.000
Internet_Protection	3	Chi-square=339.445	2	0.000
Tenure	6	Chi-square=335.372	5	0.000
Premium Support	3	Chi-square=283.528	2	0.000
Broadband	3	Chi-square=282.217	2	0.000
Payment_Type_3Grp	3	Chi-square=259.591	2	0.000
Payment_Type	3	Chi-square=259.591	2	0.000
Backup	3	Chi-square=212.116	2	0.000
Insurance	3	Chi-square=208.437	2	0.000
Monthly_Rate	4	Chi-square=168.668	3	0.000
Box_Office_Movies	3	Chi-square=142.026	2	0.000
Super_Sports	2	Chi-square=137.490	1	0.000
Average_Monthly_Bill	7	Chi-square=157.905	6	0.000
Total_Bill	3	Chi-square=134.901	2	0.000
Very_Loyal	2	Chi-square=122.198	1	0.000
Cashback	3	Chi-square=125.716	2	0.000
Family_Plus	2	Chi-square=102.654	1	0.000
Dependents	2	Chi-square=69.690	1	0.000
Partner	2	Chi-square=69.452	1	0.000
Pensioner	2	Chi-square=68.821	1	0.000
Customer_Tier	3	Chi-square=65.291	2	0.000
Landline	2	Chi-square=1.545	1	0.214
gender	2	Chi-square=0.540	1	0.462
Multiline	2	Chi-square=1.825	1	0.530

Figure 20 - Potential predictor variables that may be entered into the CHAID decision tree model ordered by statistical significance

Unless we choose otherwise, the interactive CHAID algorithm will enter the variable 'Auto_Renew' as the first predictor in the decision tree. That's because the probability value associated with the Chi-Square test indicates that its relationship with the churn variable is the *least likely* to have occurred due to random chance. In these situations, the smaller the probability value, the stronger the predictor.

Unfortunately, the table only shows the adjusted probability value displayed to three decimal places under the columns marked Adj. Prob. so we can't see the exact number. But the Chi-Square value for this variable is substantially larger than the next best predictor: the variable 'Internet_Protection' (which has the same number of categories as 'Auto Renew').

Note that the bottom three variables all have probabilities above the minimum criterion value of 0.05 for inclusion in the model. Because these probabilities are too large to be regarded as significant, these fields would never be entered at this stage using the algorithm's automatic settings.

One of the reasons I'm using an interactive modelling procedure as an example is to illustrate that different fields compete with one another to be entered into a model. What would happen for example, if we used a different random sample of data? Would the variables in Figure 20 be ordered in the exactly the same way? It's possible that they wouldn't. It's even more likely this would occur if we were using an algorithm that didn't choose predictor variables by running Chi-Square tests but instead employed some other method (as many do).

This is an important point, as many stakeholders in predictive analytics projects struggle to understand that often there is no one definitive model. When they ask the question 'What drives customer churn in our business?', it's not always possible to state that the same factors, in the same order of importance are associated with the outcome in question. What might be more reasonable, is asking the analyst to provide a list of the most commonly chosen variables and their relative ranks in terms of their overall importance in predicting churn.

Figure 21 shows what the decision tree looks like if we allow the variable 'Auto_Renew' to be the first predictor variable entered into the model.

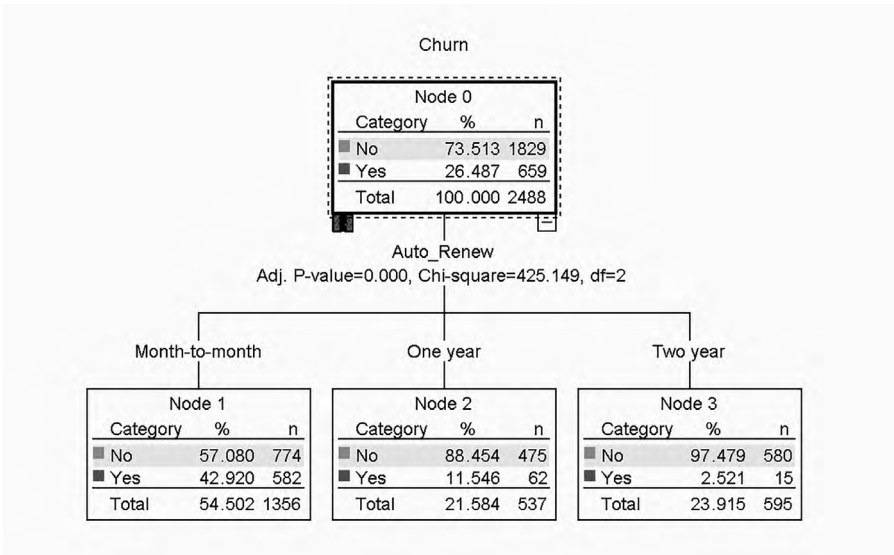


Figure 21 - The first branch of the interactive CHAID decision tree split by the variable Auto_Renew

This image reveals that the 'Auto_Renew' variable has three categories referring to how often the subscription is renewed. We can see that those who renew on a month-to-month basis are by far the most likely group to churn (42.9%). Compare this to those on a one-year subscription contract where only 11.5% churn and those on a two-year contract where a mere 2.5% cancel their contracts.

It's good advice to keep an eye on the total frequency count in each tree node as this value shows how many records fall into each respective group. Clearly the month-to-month contracts are the most popular, as this node contains 1,356 customers whereas the one-year and two-year contracts have 537 and 595 customers respectively.

What will happen if the tree grows down another level? Once again, the algorithm will check to see which predictor variable is the most useful to enter at each branch split of the current tree. By interrogating the branch showing the month-to-month contract customers in node 1, we will be able to see which predictor fields the model is considering. As Figure 22 shows, currently the variable with smallest probability, and therefore the most appropriate for entering into the model at this branch point, is the field 'Broadband'.

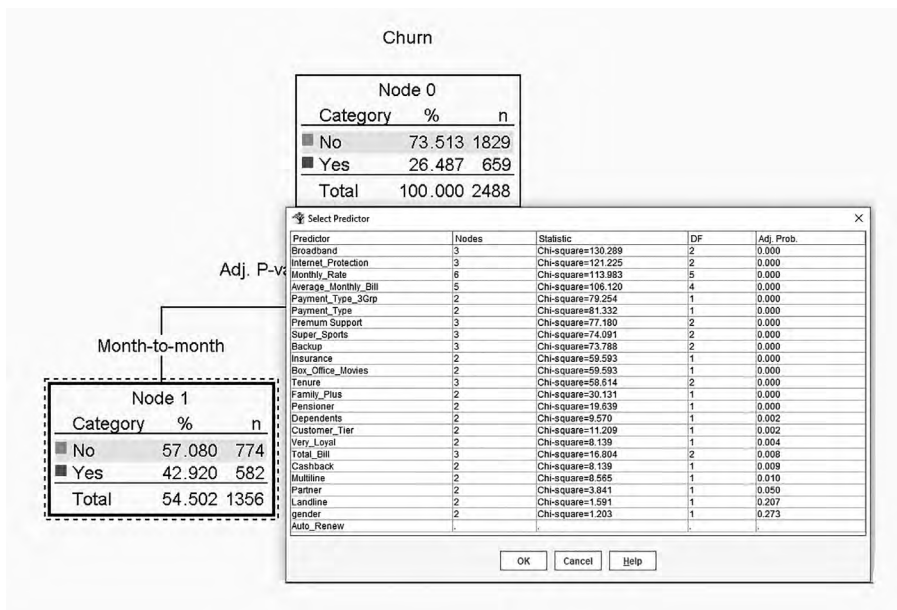


Figure 22 - Potential predictors that may be entered under the 'Month-to-month' node in the decision tree

Again, it's always possible for us to overrule the algorithm's default model-building settings, but if we allow the algorithm to choose the variable 'Broadband' at this point, the decision tree will look like the one shown in Figure 23.

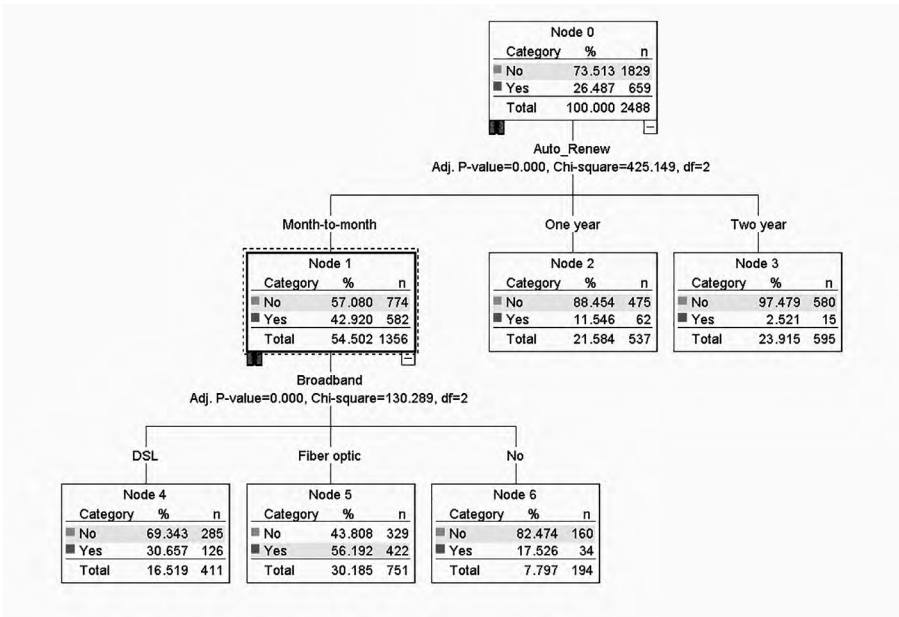


Figure 23 - The CHAID interactive decision tree grown two levels deep via the 'Month-to-month' contract branch

By growing the branch further we can see that the tree is now finding exceptions to the rule that month-to-month contracts exhibit churn rates of 42.9%. If the customers are on a DSL connection for instance, this rate falls to 30.6%, if they are on a fibre optic connection however, it rises to 56.1%, and if they don't use the broadband service it's only 17.5%.

You may recall that this partially built model is based on a random sample of 2,488 cases. Given that we expected 50% of the original 7,000 records, where are the missing 1,000 or so customers? One of the aspects of using interactive decision tree algorithms like this one, is that the procedure automatically withholds a random sample of 30% of the data so users can validate their results against a random subset. This means that the tree output we have been viewing thus far, is only based on 70% of the training sample of 3,500 records. The really useful thing is, we can request that the procedure shows us how closely the values in this tree match those when the same tree structure is applied to the validation sample. Figure 24 illustrates this.

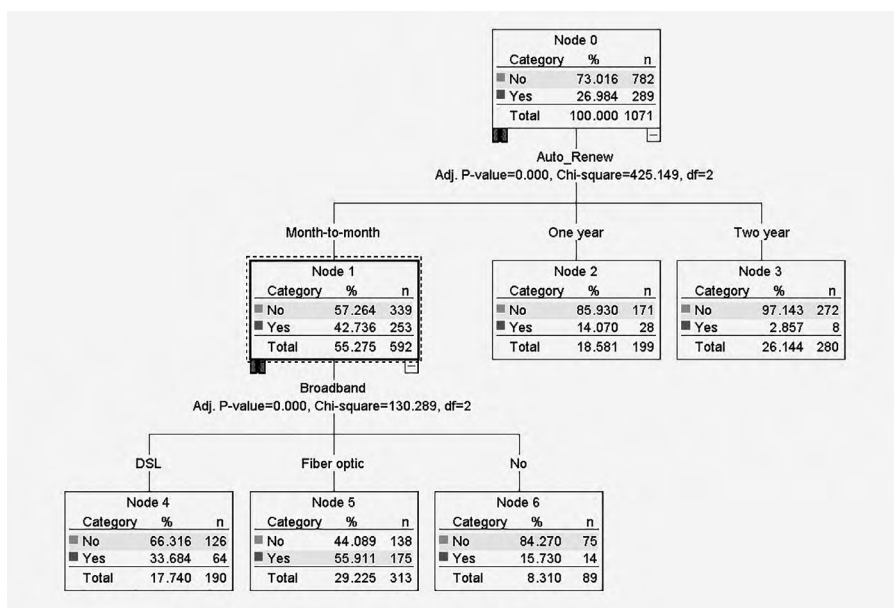


Figure 24 - The partially built decision tree model applied to a validation tree based on a withheld 30% random subset of data

The first thing to notice is that the root node in this second tree contains only 1,071 records. That's because this is the random 30% subsample of data that the interactive algorithm automatically withheld. The second thing to notice, is that the percentage values in this validation tree are different, but not by much. In fact, the percentage values in the nodes are not out by more than 3% of the values in the tree that we built in Figure 23. The point of viewing the validation tree is to estimate how stable the model is. If the model is fairly robust, it should produce similar results when it's applied to a new dataset.

Let's see what happens if we continue to grow out part of the tree. Figure 25 shows the results of allowing the branch of the model training tree for customers on month-to-month contracts with DSL connections to grow one more level deep.

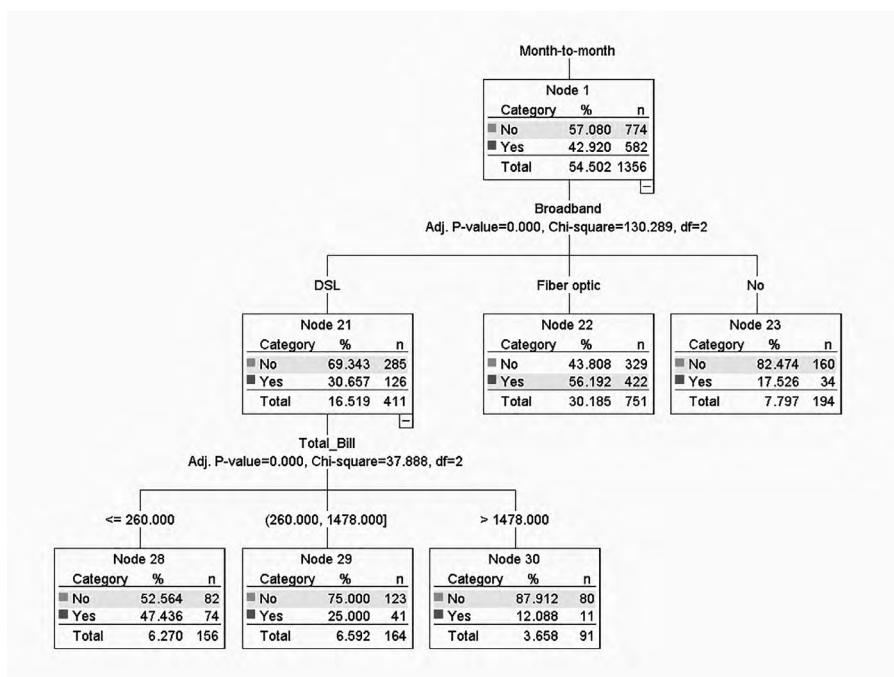


Figure 25 - The CHAID interactive decision tree grown three levels deep via the Month-to-month contract and DSL connection branch

Now the tree shows that for this segment of customers, the next most appropriate split is made by using the variable 'Total_Bill'. This has resulted in three new groups: subscribers whose total billing was £260 or less, those whose billing is between £260 and £1,478 and those with a total bill in excess of £1,478.

What we need to be aware of, is that every time the tree splits, we're creating smaller group sizes. We can see this as the number of cases in each tree node gets progressively smaller as the tree goes deeper. We all know that a sample size of 50 is not as reliable as a sample size of 500, so we should be careful when making generalisations about customer segments if we don't have a large sample of people who fall into that group.

For example, we only have 91 records in the group whose total billing was over £1,478 and only 11 of these people have churned. Furthermore, the withheld validation sample is based on only 30% of the original training data, so the equivalent node in the validation tree will have an even smaller number of cases. Should we expect to find that the same group in the validation tree also has a churn rate of close to 12%? Figure 26 shows the same branch path in the validation tree.

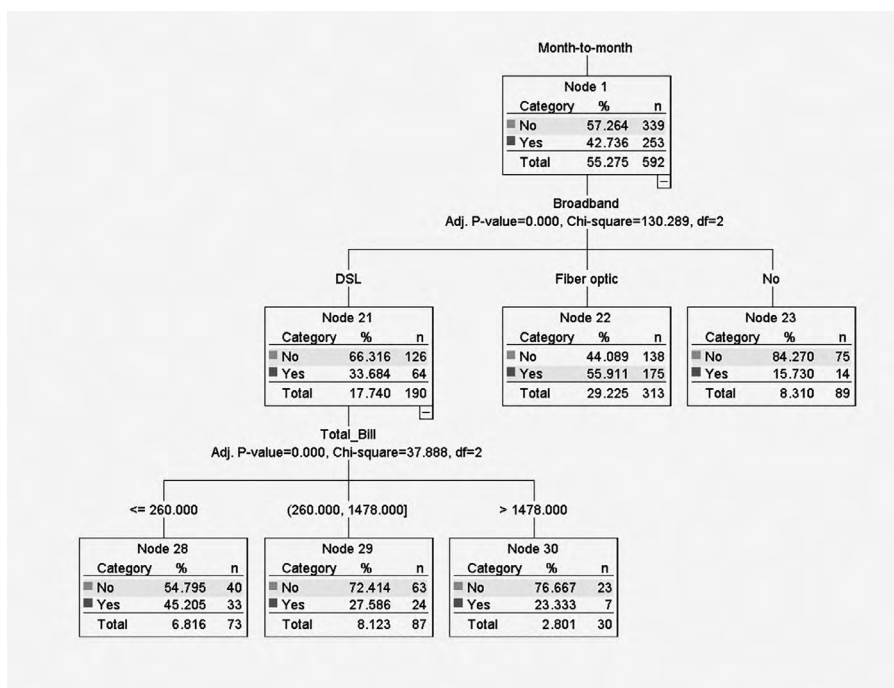


Figure 26 - The partially built decision tree model grown three levels deep but displayed in the model validation tree

The validation tree shows that Node 30 which contains the group whose total billing was over £1,478, now has a churn rate of 23.3%. This is substantially higher than the same group's 12% churn rate in the original training tree. In the validation tree, we only have 30 cases for this segment so it's not surprising that it corresponds so poorly to the results from the previous tree. Moreover, we should ask ourselves, 'Which of the two trees are more likely to accurately estimate the churn risk for this customer group?'

If anything, it's interesting to note how well the churn rates in the other groups at this branch level correspond with those of the training tree. The point of this exercise is to be aware that predictive models always make mistakes, and the process of model development means that the analyst has to check and check again until they have a thorough understanding as to where the model performs poorly and why.

One way to do this is to simply keep testing the model, perhaps with different random samples, to find out how prevalent any inaccuracy is over repeated trials. Let's assume that the model really does fit several sub-groups with less than 50 cases poorly. What might a data analyst do to address a problem like this? Here's a list of possible actions they might take:

- Do nothing - after all we shouldn't expect the model to be correct in all circumstances. As long as the overall accuracy is reasonable, we can live with a certain degree of error.
- Change the model-building criteria so that it can't produce nodes with fewer than 100 cases or is prevented from growing its branches too deep.
- Use an ensemble modelling method such as bagging (bootstrap aggregation) or random forests that build many separate trees based on different random subsamples before consolidating their various predictions.
- Use the decision tree as an exploratory method to understand the main factors that drive the target outcome and how they interact with one another. Then choose an entirely different method (whether statistical or machine learning) to build the final model.

Suppose we opt for the first option. We ignore the issue with inaccurate model predictions for small customer segments (by the way, this problem is directly related to the issue of *overfitting* that we mentioned earlier). I should also point out that we don't have to build a decision tree model via the interactive mode that we've been using thus far. Instead we can ask that the entire decision tree model is automatically built, using the default settings, against the full training sample of 3,359 records. Unfortunately, the final tree is too complex to show all the details in a single image, but Figure 27 displays a 'zoomed-out' view, so you have a sense of its dimensions.

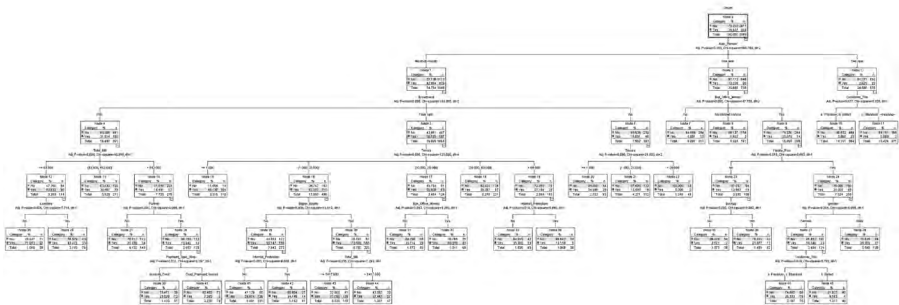


Figure 27 - The complete CHAID decision tree based on the entire training sample of 3,359 records

Although we can't view the final tree in detail here, the IBM SPSS Modeler application does provide a handy predictor importance chart (Figure 28) showing which variables were used to create the decision tree and their relative importance to the overall model accuracy. This is calculated based on a separate analysis procedure that measures how much the model's accuracy is reduced if the variable in question

is removed.

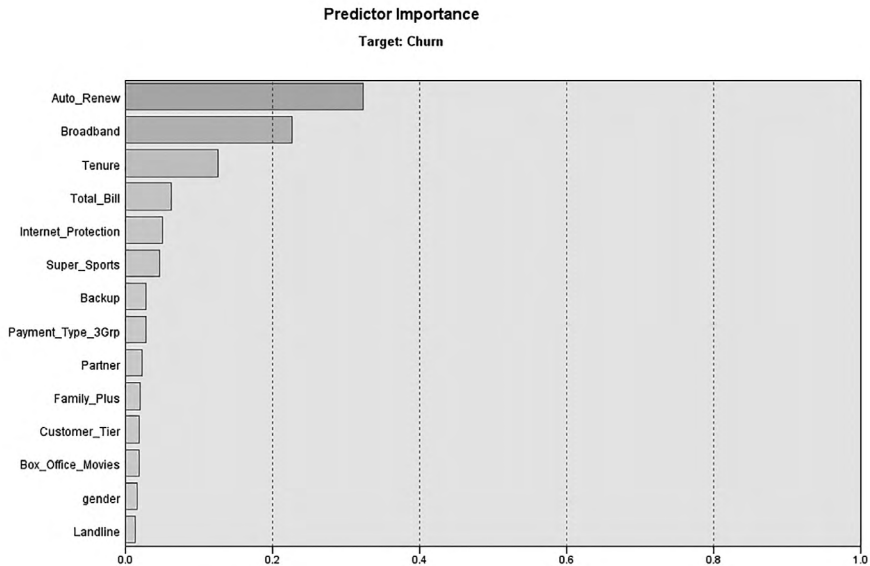


Figure 28 - Predictor importance chart showing the relative impact of the included decision tree variables on model accuracy (larger bars indicate greater importance)

I’ve deliberately chosen to introduce the topic of modelling using an interactive decision tree example simply because this rule-based method of predicting outcomes is one of the easier techniques to understand. There are several different decision tree algorithms available today. As mentioned earlier, the CART (classification and regression trees) algorithm developed by Breiman and Friedman (1984) is an alternative to the CHAID method, as is the C5 decision tree program developed by Ross Quinlan (1986). However, the traditional approach to predicting the outcomes of a categorical target variable is to use a statistical method.

Logistic regression is by far one of the most popular and well-used predictive modelling techniques available. In fact, its usage can be traced back to the 19th century when the logistic function was developed as a model of population growth. Unlike a decision tree, logistic regression does not produce visual output in the form of a hierarchy of decision rules, but instead provides the analyst with a series of values that form a predictive equation.

Using these values, the analyst can calculate the probability that a row of data belongs to a specific dependent (or target) variable category. The values themselves are referred to as the model coefficients, and they create a regression equation exactly like the equation for a straight line we encountered in the previous chapter. In that particular example, we saw that we could derive a simple formula allowing us

to estimate a car's horsepower by multiplying its engine size by 0.359 and adding a constant of 35.75. This means that after allowing for a minimum of 35.75, every cubic inch of engine size is worth an increase of 0.359 in horsepower.

Unfortunately, the coefficient values that logistic regression produces cannot be directly interpreted in the same way as linear regression coefficients. With this kind of regression we're not trying to predict a continuous outcome, such as a test score or a temperature value, but rather the probability of belonging to a category group.

To do this, the coefficients that logistic regression calculates are in fact based on the logarithm of the odds that indicate how much the values of an independent variable affect the probability of belonging to a specified reference category in the dependent target variable. Clear? If not, you won't have been the first person to fall foul of the rather involved mathematics that underpin logistic regression.

Suffice to say that a special formula (known as a link function) converts the results of the logistic regression equation into probabilities. The probabilities can in turn be used to predict the outcomes in the target variable. Statisticians are well-used to poring over tables of coefficient values like those displayed in the logistic regression output in Figure 29.

Variables in the Equation ^a						
	B	S.E.	Wald	df	Sig.	Exp(B)
Step 8 ^b Tenure	-.028	.003	83.937	1	.000	.973
Partner(1)	.082	.099	.693	1	.405	1.086
Broadband			107.781	2	.000	
Broadband(1)	.785	.215	13.315	1	.000	2.192
Broadband(2)	1.759	.224	61.636	1	.000	5.805
Internet_Protection			16.684	1	.000	
Internet_Protection(1)	.473	.116	16.684	1	.000	1.606
Premium_Support			4.265	1	.039	
Premium_Support(1)	.241	.117	4.265	1	.039	1.272
Box_Office_Movies			9.946	1	.002	
Box_Office_Movies(1)	-.327	.104	9.946	1	.002	.721
Auto_Renew			49.057	2	.000	
Auto_Renew(1)	1.554	.249	38.976	1	.000	4.732
Auto_Renew(2)	.843	.249	11.461	1	.001	2.322
Payment_Type			18.645	4	.001	
Payment_Type(1)	.394	.133	8.767	1	.003	1.482
Payment_Type(2)	.183	.403	.206	1	.650	1.200
Payment_Type(3)	.019	.163	.014	1	.905	1.020
Payment_Type(4)	-.080	.162	.240	1	.624	.924
Constant	-3.176	.290	119.896	1	.000	.042

Figure 29 - Coefficients and model terms table generated by running a logistic regression procedure

The purpose of showing this image is to reveal how much traditional statistical approaches differ from methods like decision trees. In Figure 29, the values in the column marked B are the log-odds units that we just discussed. The column headed S.E shows the standard errors of the model terms (the B values). These allow the analyst to estimate how reliable the coefficients are. The columns headed Wald and df refer to the Wald significance test and its accompanying degrees of freedom. The column headed Sig. shows probability values resulting from the Wald significance test. These provide guidance as to whether or not the model term is making a genuine contribution to the model fit. Values above 0.05 would indicate that the term was so weak it isn't really helping to predict the outcome. Finally, the Exp(B) refers to the exponentiated version of the B values in the first column. These values are known as odds ratios and they are used to assess how much a change in the values of the independent variable increase or decrease the odds of being in reference outcome category (i.e. in this case, the risk of churning).

For comparison with the decision tree model we saw earlier, Figure 30 shows the predictor importance chart for the logistic regression model we have just looked at. You can see that compared to our previous example, the logistic regression model appears to have about five predictor fields that don't contribute much to the model accuracy.

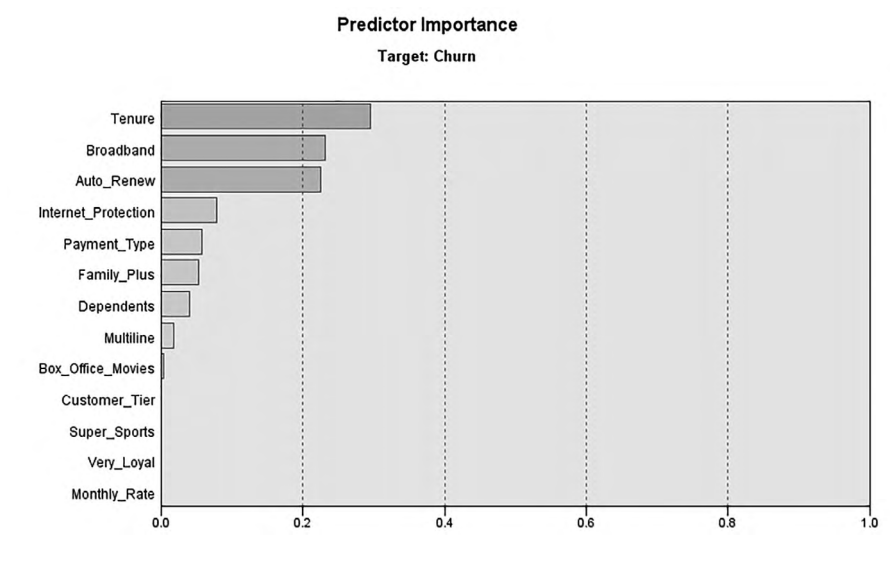


Figure 30 - Predictor importance chart showing the relative impact of the included logistic regression variables on model accuracy

Before we finish this chapter, let's look at one further example of a modelling technique. This time we will use an 'old fashioned' machine learning method: a neural network. Neural networks are often likened to the functioning of a brain.

This is because they're composed of a number of interconnected layers of artificial neurons.

As a family of techniques, they are employed for applications such as clustering data, predicting category or continuous outcomes as well as specialised tasks like image or speech recognition. Neural networks can sometimes perform impressively in predictive modelling when compared to more traditional approaches, but it's important not to be too over-awed by them.

In machine learning, a basic neural network uses a set of predictor variables as the input layer that is entered into the algorithm. So, four continuous variables (e.g. age, tenure, average spend and number of children) used to predict an outcome like contract renewal, would act as the four factors in the input layer.

The critical aspect of a neural network, however, is how this input layer is then used to create at least one hidden layer. The hidden layer is normally comprised of a set of network neurons. To understand a neuron, think of how the logistic regression model we just saw, has a number of B coefficients that can be combined to form a regression equation. In the regression equation, each predictor variable is multiplied by its respective coefficient and a constant is added to generate a single value for a case. This value is then converted via the logistic link function into a probability which represents the prediction itself. It's possible to generate a neural network that does just this and has only one neuron in its hidden layer. Figure 31 illustrates this, using the same dataset we have been working with so far.

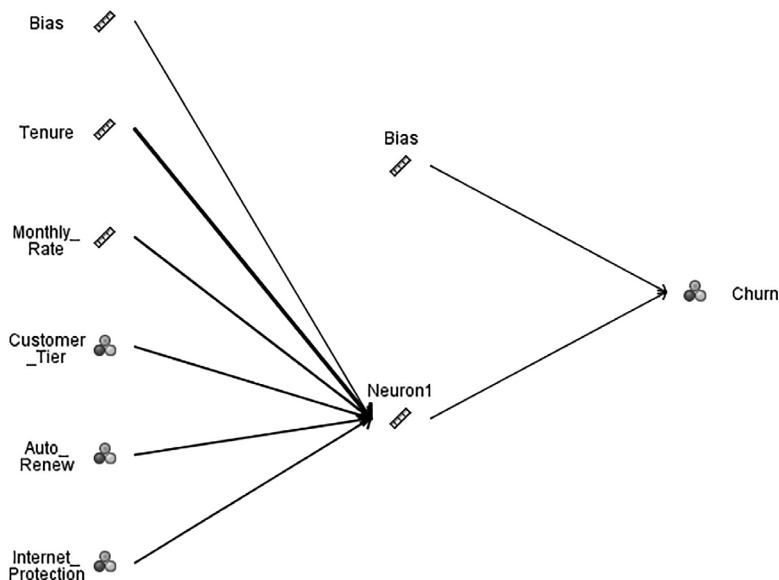


Figure 31 - Output diagram showing a Neural Network model with a single neuron in a hidden layer

Figure 31 illustrates the structure of a neural network model where several input variables have been combined in a single neuron. This is analogous to combining variables in a regression formula. The network even has a constant value just like the intercept in a regression model. In this context, the model constant is referred to as the bias.

You may notice that the thickness of the lines in the network diagram varies slightly. The connections between the predictor variables and the neuron represent the coefficients in the model but again, in this context, these are referred to as the model weights. The weight value for the variable 'Tenure' happens to be 0.769.

The algorithm finds these weights by beginning the model-building process with a random set of values and evaluating how accurately they predict the outcome in question. It repeatedly updates the weights trying to find the optimum combination of weight values that will maximise the predictive accuracy. The process is a little like turning the dial on an analogue radio to tune into a station. In fact, one of the problems with neural networks is that they can predict the outcome too accurately on the training sample dataset by producing an overfitted model that won't work well on alternative data samples.

To prevent this, just like in our example using interactive CHAID, the model withholds a random subset of 30% of the data so that the algorithm never sees all the data and is so is less likely to produce an overfitted model. The model shown Figure 31 is based on a fairly standard type of neural network known as a multilayer perceptron. This particular example is somewhat unusual in that it only has one neuron.

If we re-run the procedure using the algorithm's standard automatic settings, it will generate a model comprised of multiple neurons. Again, the algorithm tries out different solutions using a different number of neurons. We can think of the use of multiple neurons as an attempt to utilise the variation in different combinations of the input variables. In such neural network structures, every variable contributes at least something to each neuron, but certain variables have larger weight values associated with certain neurons in the hidden layer.

Figure 32 shows the results of running the procedure again but this time allowing the algorithm to choose the number of neurons in the hidden layer. The result is a model that is comprised of seven neurons. It's interesting to note that, based on an initial test, this network was only marginally more accurate than the one using a single neuron.

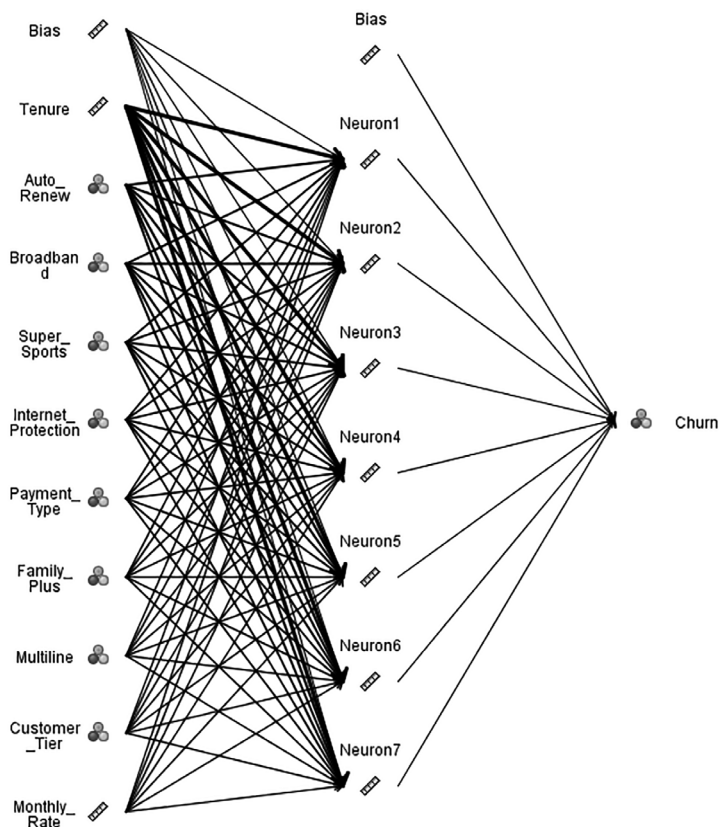


Figure 32 - Output diagram showing a Neural Network model with multiple neurons in a hidden layer

I should also point out that the I've developed this model using the same variables that the earlier CHAID model selected. That's because this particular neural network algorithm doesn't attempt to evaluate whether a predictor variable can make a meaningful contribution to predicting the outcomes in a target variable. In other words, I could include a field that records the customers' zodiac star signs and it would attempt to use it. Although variables that don't help predict the outcome are unlikely to have large weight values as they won't be contributing much to the model, they nevertheless clutter it up and may add unnecessary complexity. For that reason, analysts take care when choosing the relevant predictor fields to use with neural networks.

To further compare the model in Figure 32 with the previous modelling approaches, Figure 33 shows the predictor importance chart for this neural network.

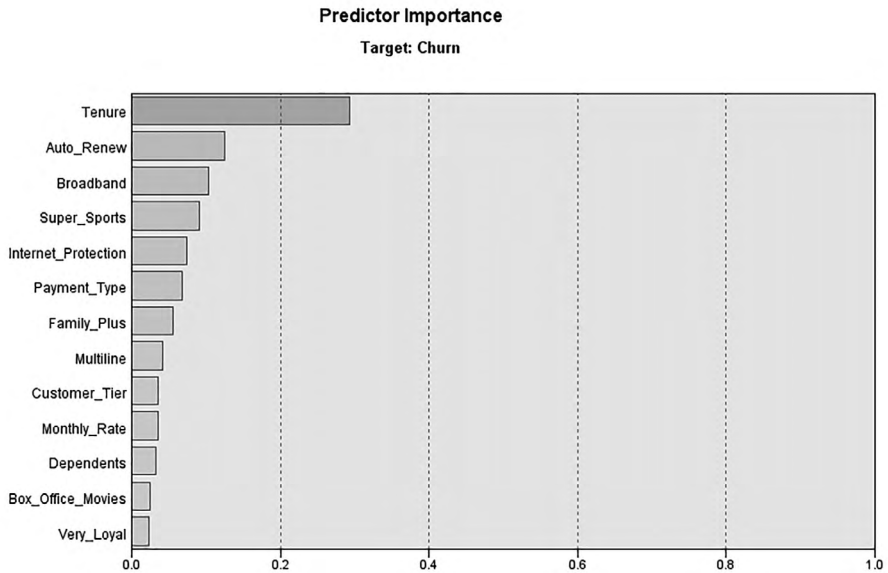


Figure 33 - Predictor importance chart showing the relative impact of the Neural Network variables on model accuracy

Finally, you should bear in mind that, not only do many neural network models contain a lot more than seven neurons, they can also contain multiple hidden layers. Given that the output generated by these models is fairly opaque and their structures can sometimes be tremendously complex, it's little wonder that they are often referred to as 'black box' techniques.

Having spent some time briefly looking at how predictive models are created, in the next chapter we will investigate the various ways in which we can evaluate their worth.

CHAPTER 6

WHAT DOES 'GOOD' LOOK LIKE?

As we have already seen, the CRISP-DM process contains a step for assessing the results of the modelling phase: evaluation. Figure 34 reminds us that the work carried out during the business understanding phase provides the essential context in which we evaluate any model produced during a predictive analytics project.

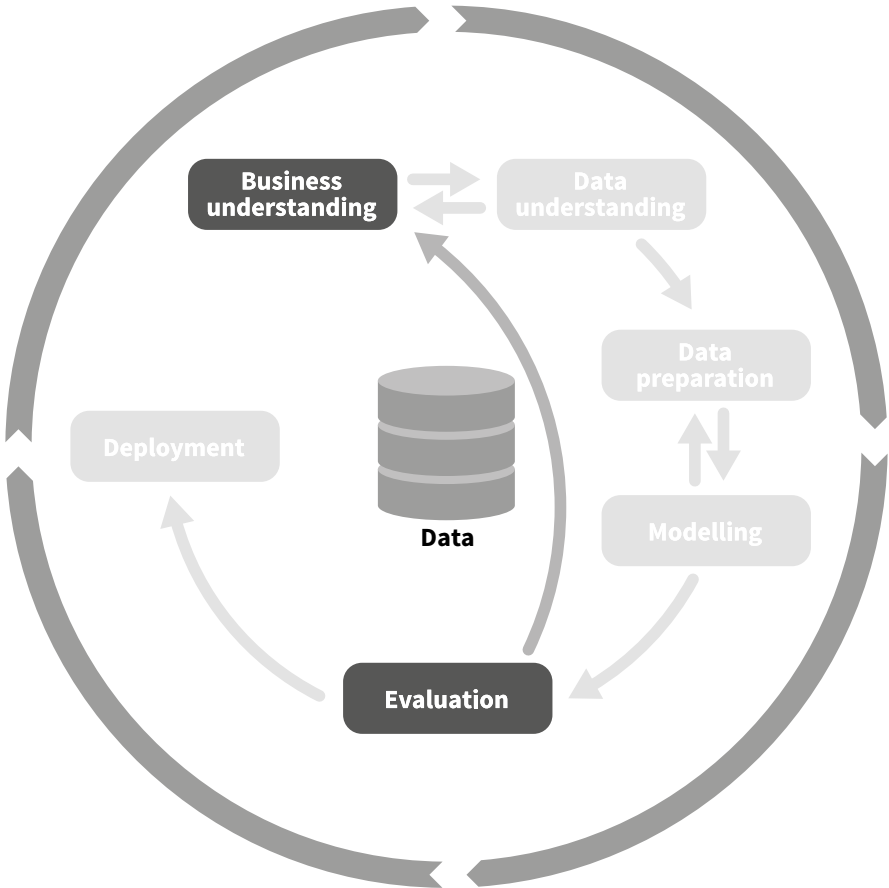


Figure 34 - The CRISP-DM methodology showing a direct link between the evaluation phase and the business understanding phase

This is important as it's easy to get side-tracked when trying out different modeling solutions. The project team need to be fully focussed on the success criteria they established during the first few days of the initiative. Without these criteria,

analysts can waste precious time assessing model performance without a clear idea as to what actually constitutes a useful or valuable result.

It's vital to understand that what might be regarded as an excellent model in one set of circumstances can also be regarded as completely inadequate in another. For a start, predictive models always generate some errors when attempting to estimate a given outcome. In fact, even models with close to 100% accuracy are usually badly flawed due to overfitting or the presence of information that wouldn't be available until the true outcome was already known.

An example of this latter situation occurred in a project I was involved in. The team I was working with were attempting to build an application on behalf of a bank. The point of the application was to accurately estimate the likelihood that the bank's mortgage customers would switch to a new provider. However, the initial results indicated suspiciously high levels of accuracy when predicting this outcome.

A further investigation of the model showed that the accuracy was greatly increased by the presence of a particular variable that recorded the kind of communications the bank had sent to each mortgage customer. Each category code in the communication variable related to things like reminder emails, marketing collateral and letters acknowledging changes in the customers' circumstances.

One of these codes was especially important with regard to the model accuracy. It turned out that it related to a letter that in essence stated, 'We're sorry to see you've switched your mortgage to another provider'. In other words, this was information that could only be known after the event in question had occurred. These situations are known as self-fulfilling prophecies and they are not uncommon during the initial model evaluation phase.

Nevertheless, a key goal of the modelling process is of course to try to minimise such errors. But even when trying to predict a two-category outcome, we can't always assume the level of accuracy in both outcomes will be the same. For instance, just because a model is 85% accurate overall, doesn't mean that it is 85% accurate in predicting both outcomes. It may well predict every record to have the same value (e.g. remaining a current customer) and if 85% of the data happens to have that outcome, then the overall accuracy is 85%.

A colleague worked on a project predicting the likelihood of particular types of tweets getting retweeted. The first round of modelling indeed produced a model that was able to correctly identify which Twitter posts would be retweeted with incredible accuracy. Initial jubilation was tempered when the analyst realised that in fact 80% of all the tweets in the sample were retweeted and that the model achieved its apparently astonishing accuracy simply by predicting that every post would be retweeted.

These situations are related to the notion of false positives and false negatives. When attempting to predict category outcomes such as customer churn, response to an

offer, readmission to hospital or the failure of a machine or asset, there is usually an implied event that we are trying to detect. This event can take the form of a customer cancelling their contract or a patient being readmitted.

When we are assessing how well the model performed against our training sample, it's highly likely that, for some records at least, it will have predicted that this event had occurred, when in fact it hadn't. This is known as a false positive. Conversely, it probably also predicted that for some records in the sample, the event didn't occur when in fact it did. These outcomes are known as false negatives.

There are likely to be situations where minimising false positives is more important than false negatives, and vice versa. If you're trying to develop a model that detects a fatal illness, you'll probably aim for a model that has as few false negatives as possible, because a false negative would mean that the model failed to detect the disease when the patient actually had it. With such serious consequences we tend to err on the side of caution, so this situation could be regarded as more dangerous than predicting that the patient probably has the disease when in fact they don't (a false positive).

The key to evaluating these kinds of classification models is understanding the costs associated with the false positives and false negatives just as much as understanding the gains that can be made by correctly predicting the outcome. Despite how obvious this is, it's one of the most common reasons that initiatives driven by predictive analytics fail to make an impact in the real world.

Time and again, projects are commissioned without adequately stipulating the success criteria. Simply stating that the project should aim to build a model that accurately predicts an outcome is not nearly enough information to stand a reasonable chance of success. Later in this chapter we will look at how analysts can assess models by taking into consideration the costs of predictive errors and the benefits of accuracy. Before we get there though, let's take a brief look at the key factors stakeholders may need to consider when aiming to develop a good predictive model.

6.1 Accuracy

It goes without saying that only predictive models that are able to predict an outcome sufficiently accurately can be regarded as useful. But what we mean by sufficiently accurate depends on the context in which the model is to be used. If we want to predict an outcome that only occurs 1% of the time, then technically speaking, any model with an accuracy of greater than 1% at predicting that particular outcome may be seen as an improvement. For this reason, it is essential that analysts establish a baseline against which to judge the model.

As we have already seen, there are costs associated with false positives and false negatives and we must be aware of these when assessing model accuracy. Within statistics and predictive analytics, there are a host of specific metrics and charts that

are used to measure accuracy and model fit. Unfortunately, this means that because there is no single universal measure, it's easy to develop a model that exhibits the highest accuracy using one criterion and the worst using another. So care must be taken to choose criteria that help us select a model that is both accurate and useful.

6.2 Interpretability

To say that there are many different predictive modelling algorithms available to analysts these days is something of an understatement. Some approaches, like the neural network technique we looked at in the last chapter, may yield so-called black box models. These are models that either can't be directly interpreted in the same way that certain statistical or rule-based models can, or that are so complex that making sense of them is extremely difficult.

In some fields model accuracy can be seen as more significant than interpretability, so these algorithms may be regularly employed. In other disciplines however (such as credit scoring, epidemiology or social research), being able to understand how the model generated its predictions is of paramount importance. It's fair to say that, in an ideal world, most analysts would prefer to create models that are highly accurate and easily interpretable.

6.3 Stability

Analytical models are based on samples of data collected under certain circumstances and within specific timeframes. We should not be surprised to find that a model fails to generate accurate results when applied to a range of values or circumstances that are very different from the ones it was developed under. For example, model accuracy may decay over time as changes in fashion, demographics, competitor behaviour or market offerings proliferate from the time period in which the model was developed.

Also, if the model is based on an unrepresentative sample, we can find that it generates inaccurate predictions or even wild estimations when encountering an unfamiliar case. Even certain relatively novel combinations of demographic factors such as ethnicity, gender, age and region can mean that the model is unable to accurately predict the outcome of interest. Stable models, however, are able to generate reliable predictions and estimates with a wide range of data combinations over a useful period of time before they need to be updated or refreshed with new data.

6.4 Coherence

As mentioned earlier, many analysts only work with models that can be directly interpreted. One of the reasons for this is to make sure that the model makes sense. It is not unusual to discover that a model uses counterintuitive rules or nonsensical relationships to estimate a value. Examples include price rises increasing the likelihood to purchase, missing values for variables such as age generating higher

estimates of revenue or low satisfaction scores reducing likelihood to churn.

Of course, there may be a sensible explanation for these contradictory relationships. However, more often than not, what is really driving the relationship is a hidden variable that explains what's going on. Perhaps price rises increase the likelihood to purchase because they are related to higher demand in the market. Therefore, demand is the driving factor and price is merely a function of it.

Missing data for variables such as age may indicate that the person registered for a product or service through a different channel (e.g. in-person as opposed to via the website) and that in reality, the channel is the key predictor. Even lower satisfaction scores could simply indicate that a person cares more about a service or has previously complained and subsequently received a discount so lowering their likelihood to defect. Coherent models are valued not only because of the obvious insights they deliver, but because they provide reassurance that the model is not based on a combination of spurious relationships.

6.5 Simplicity

Most predictive models are multivariate in nature. This means they are developed from a combination of interrelationships between multiple variables or model terms. Many of them can be likened to a house of cards where each layer is precariously added to a previous tier. Complexity therefore not only leads to issues with interpretability but also stability. For these reasons alone, simplicity is a key criterion when selecting a predictive model.

For example, an analyst might well reject a model with an overall accuracy of 85% based on 18 variables in favour of one with an accuracy of 82% but only based on 8 variables. This is because including the terms from an additional 10 variables to gain a mere 3% of accuracy may be regarded as poor trade-off. Statisticians refer to this principle as parsimony, implying that the frugality of a model is to be commended.

6.6 Performance

A final consideration is the computational performance of the model. Different modelling algorithms use resources in different ways. Some might not work well with categorical data and require data transformations to perform effectively. Some might take a very long time to build a final model or require significant memory allocation and processing power. These requirements can mean that it takes much longer to refine and uncover a final satisfactory model. Likewise, when the model is deployed to generate predictions, it may require unacceptable resources in terms of computing power, coding effort and time to score new cases.

The most notorious example of this occurred in 2009 when Netflix awarded a \$1 million prize to a consortium of analysts. The team of analysts had taken part in a Netflix-sponsored competition where the goal was to develop a more accurate movie recommendation engine for the company's subscribers. Although the winning team of analysts increased the recommendation accuracy by the required threshold of

10%, Netflix had already significantly improved the accuracy using an earlier model. It was felt that the amount of engineering required to implement the complex code from the winning model, outweighed the gains that might have been achieved from a further increase in accuracy (Johnson, 2012).

As Figure 35 below shows, the first key sub-task associated with the evaluation phase is to assess the results of the modelling phase.

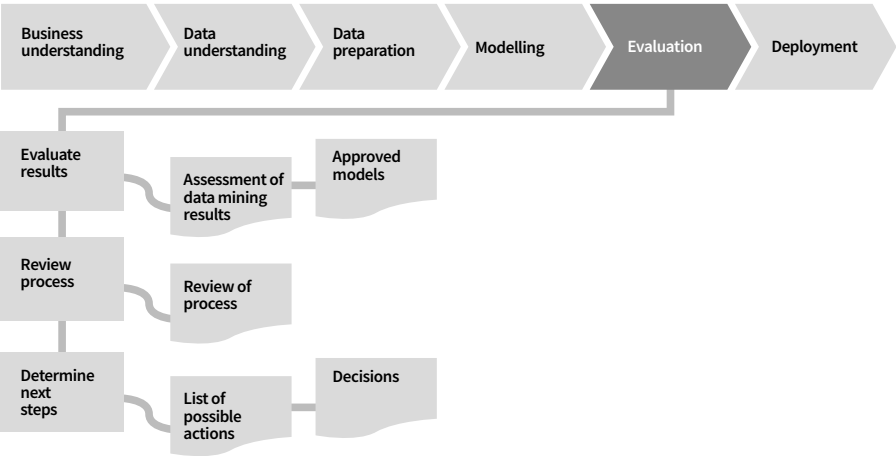


Figure 35 - Breakdown of the evaluation phase into tasks and sub-tasks

Assessing models purely in terms of their accuracy is surprisingly difficult. This may come as surprise to some. Surely, all we have to do is compare the predictions that a model generates to actual outcomes. The problem here is what is what is meant by accuracy. As we’ve already discussed, accuracy implies benefits just as inaccuracy implies costs. This is easy to understand when the example is betting on a horse, but less so when the outcome in question relates to whether or not an employee will leave their job, the diagnosis of a patient or the failure of an industrial pump.

Figure 36 illustrates exactly this issue. It shows three tables that measure the accuracy of three different models that are attempting to predict churn in a sample of 3,574 customers. In the first table (Model A), the model correctly classifies 89.3% of the customers who have not churned as current customers. However, it only classifies 51% of the customers who defected as churners.

In the second table (Model B), the accuracy in predicting current customers has dropped to 81.6%, but the accuracy in predicting churners has increased to 68.2%. The last table (Model C) shows a further decrease in accurately classifying current customers (73%) but a reciprocal increase in the accuracy of detecting churners (77.8%). The question the analyst faces is which model is the best? Without additional information, it’s impossible to say.

Model A: Actual Churn by Predicted Churn				
No			Churn_Predicted	
Yes				
Churn_Actual	No	Frequency	2330	278
		Percent Correct	89.3%	10.7%
	Yes	Frequency	473	493
		Percent Correct	49.0%	51.0%

Model B: Actual Churn by Predicted Churn				
No			Churn_Predicted	
Yes				
Churn_Actual	No	Frequency	2129	479
		Percent Correct	81.6%	18.4%
	Yes	Frequency	307	659
		Percent Correct	31.8%	68.2%

Model C: Actual Churn by Predicted Churn				
No			Churn_Predicted	
Yes				
Churn_Actual	No	Frequency	1919	689
		Percent Correct	73.6%	26.4%
	Yes	Frequency	214	752
		Percent Correct	22.2%	77.8%

Figure 36 - Comparing the accuracy of three models using tables

In each of the tables we have focused on how accurately the model predicts the two outcomes i.e. the benefits. We haven't explicitly looked at the costs associated with those situations where the model has mis-predicted the outcome. If we were able to estimate, even roughly, how much revenue we lose when a customer churns, how much it costs to retain a customer that we've decided is at risk of defecting (e.g. with a discount or a proactive offer) and how much revenue we gain from satisfied current customers who have little risk of churning, then we would be in a much stronger position to choose the best model.

Another thing to bear in mind is how the three models decide to assign a customer to the 'Yes' or 'No' groups when predicting churn outcome. They did this by computing the probability of each case being in one of those two groups. For example, if a model predicts a customer to have a probability of 0.23 (23%) of being in the 'Yes' group (i.e. they will churn) then it will assign them to the 'No' group (i.e. a current customer). This is simply because a 23% chance means that, on balance, they probably won't fall into the churn category.

Obviously, if the model predicts a customer to have a probability to 0.62 (62%) chance of being a churner, it will assign them to the ‘Yes’ group. In other words, a model of this type simply assigns cases to one group or another based on whether or not the probability of being in that group is greater or less than 50%.

Looking at the tables in Figure 36, for all we know the probability could have been 0.505 (or 50.5%) in every case where the customer churned. Conversely, in every case where the customer was predicted to be a current customer, the probability might have been 0.495 (49.5%). The tables don’t show us the probability values that the models’ calculated for each customer. They simply summarise the model accuracy by assigning cases to their predicted categories based on the rough, rule-of-thumb that if their chances of being a churner exceed 50%, then they are assigned to the ‘Yes’ category. For this reason, analysts often use special charts to visualise model performance.

6.7 Visualising model performance

One simple method to view how accurately a model can predict a specific outcome is with a gains chart. The purpose of the gains chart is to show the proportion of records in a target group that the we can ‘gain’ by using the model. Perhaps the easiest way to make sense of it is to note that the chart itself can only focus on one group within the target field at a time.

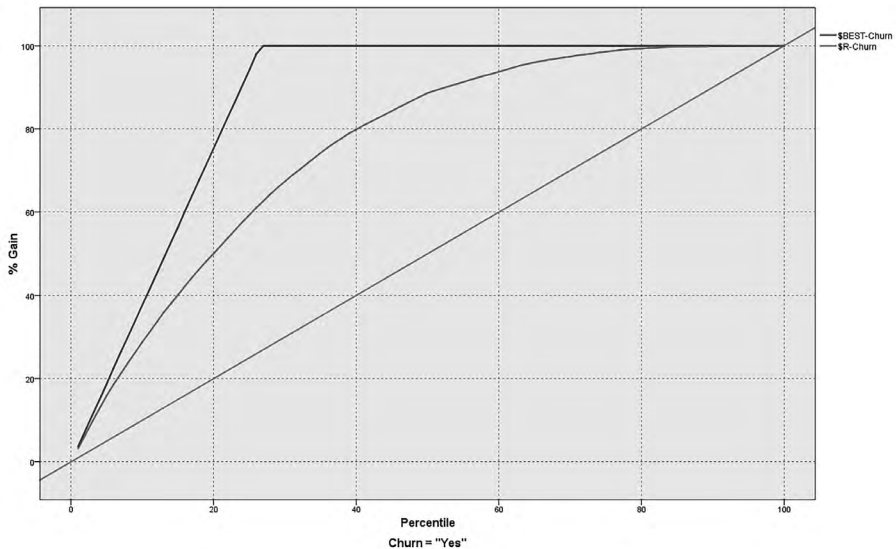


Figure 37 - Gains chart showing model performance compared to a random and a ‘perfect’ model

Take a look at Figure 37. The group of interest are the customers who have churned (as evidenced by the label on the horizontal axis indicating that Churn= ‘Yes’). This axis shows what percent of the entire dataset is under consideration. The vertical

axis represents the percentage of churners the model is gaining.

The chart itself contains three lines. The lowest line in the chart is the straight diagonal one that simply indicates what proportion of churners we might expect to find if we were to randomly sample the data. This line is for reference only and allows us to see how well our model performs compared to it. The diagonal line always looks the same. It is perfectly proportionate to the vertical axis as it shows that if we just adopted a random approach, we could only expect to find (or gain) 20% of the customers in the churn group from 20% of the data, and 50% of the churners by sampling 50% of the file. This is the equivalent of guessing.

We would always expect a reasonable model to perform better than random. If you look at the top line in the chart, we can see what the model performance would look like if we could predict outcome perfectly. This line is marked \$BEST-Churn in the legend as it represents the best possible model. If you look at where this line levels off, it indicates that we could gain 100% of the churners (vertical axis) from 27% of the data (horizontal axis). This is simply because 27% of the records in the dataset belong to customers who have churned.

Having established what a random model and a perfect model would look like, the middle curving line shows us how many customers we might expect to find using the predictive model itself or, alternatively, how much better the model is than random and how much worse it is than perfect.

This line is marked as \$R-Churn in the legend and it happens to refer to the prediction field that a CHAID decision tree generates in IBM SPSS Modeler. Gains charts are particularly useful for people working in marketing applications. In those situations, it may not be feasible to contact all of the customers in a predicted target group. The gains chart sorts the predictions in order of their probabilities so that the analyst can choose those customers where the model has the highest degree of confidence.

To that end, within the SPSS Modeler application, moving the cursor along any of these lines causes a pop-up label to appear telling us what proportion of churners we can gain as we increase our sample size. Figure 38 shows that the model indicates we could expect to find about 63% of the people who churned from only 27% of the data. Indeed, we can directly select this group using the chart. This approach is very useful because if we only have the resources to contact 10% of the customer base then the chart will tell us what proportion of our target group we can expect to gain by selecting the 10% of cases where the model is most confident.

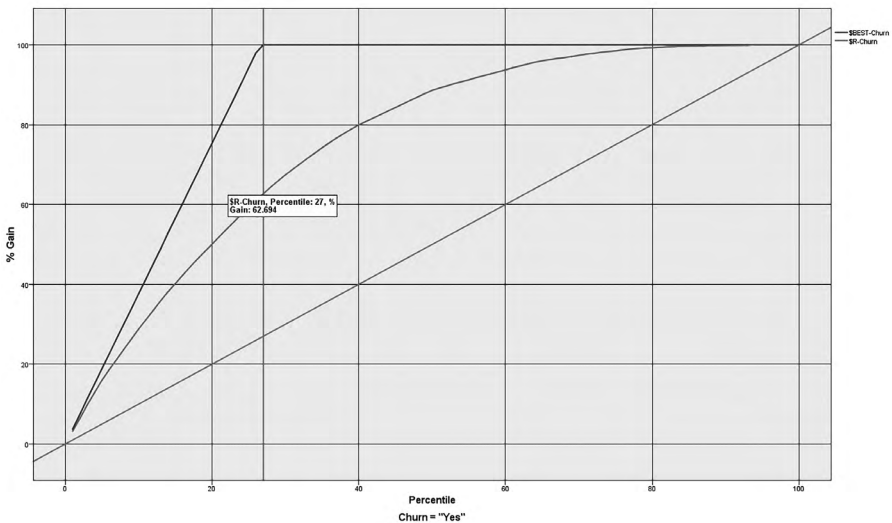


Figure 38 - Gains chart showing the proportion of churners detected by selecting the top 27% of the data in terms of the model confidence

Charts like these are also very useful for comparing model performance. Figure 39 below shows how this can be done using a gains chart. The middle two lines show the performance of two separate models. The lighter curving line marked as \$C-Churn is a model generated by a C5 decision tree and we can compare how it performs to our earlier CHAID model (\$R-Churn) as indicated by the slightly darker curving line.

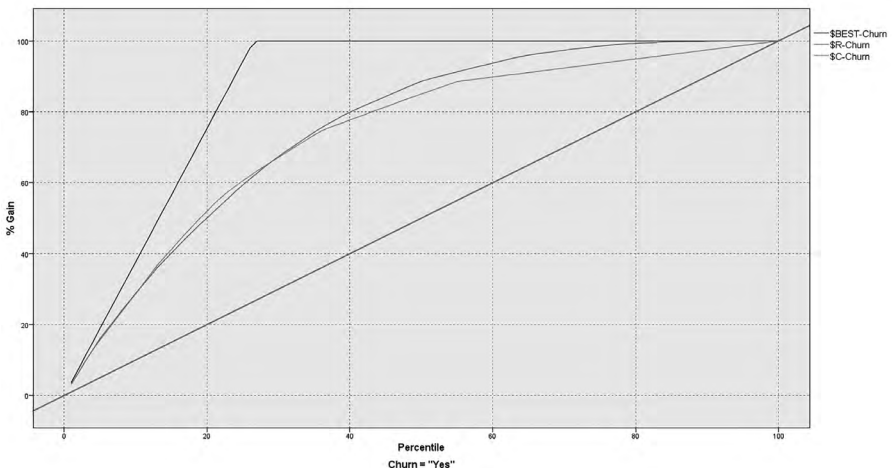


Figure 39 - Gains chart comparing performance for two predictive models

Note that around the 20th percentile on the horizontal axis, the lighter line from the C5 model is slightly above (barely) the darker line indicating the CHAID model performance. Whereas around the 50th percentile the situation is reversed. This tells us that, all else being equal, if we wished to select the best 20% of the predictions that detect churn, we should choose the C5 model (\$C-Churn) since it detects a slightly higher proportion of churners at that point than the CHAID model (\$R-Churn). If, on the other hand, we wished to select the best 50% of predictions, we should choose the CHAID model since at the 50th percentile it clearly outperforms the C5 model.

Another kind of model performance chart is known as a receiver operating characteristic curve (or ROC chart). This is a slightly more technical chart and I include it here as an example mainly due to its widespread usage throughout statistics and data science. Earlier, we discussed the notion of predicting something to be true when in actual fact it isn't - a false positive. Of course, the converse of this is when a model predicts something isn't true when in fact it is - a false negative. ROC curves are an attempt to show the relationship between the rate at which a model accurately predicts the true outcome (the true positive) and the rate at which it generates false positives.

For the uninitiated, the business of comparing false positive outcomes against true positive outcomes can be confusing (it's also confusing for a lot of analysts!) Nevertheless, I shall attempt to at least explain what some of the terminology associated with this approach refers to. The first thing to note is that in statistics, the true positive (TP) rate is known as the model sensitivity.

Sensitivity is like the gain value we saw in the previous charts. It reflects the ability of the model to correctly detect the occurrence of the event in question (e.g. customer churn or patient readmission). The sensitivity (true positive) rate is normally shown on the vertical axis of a ROC chart.

A true negative refers to the situation when the model correctly predicts a non-event (e.g. the customer does not churn). The rate at which a model correctly predicts true negatives is known as its specificity. However, ROC charts instead display the False Positive (FP) rate on the horizontal axis, which is calculated as 1-specificity. Think of this axis as measuring the proportion of cases that turn out to be false alarms. As you move along the ROC curve, you get more true positives but at the expense of encountering more false positives.

Figure 40 shows a ROC curve for the CHAID and C5 decision tree models we evaluated earlier. Once again, the straight diagonal line indicates what the model would look like if it was random. Any curve above this line indicates an improvement on a random model. Focusing on the CHAID model (\$R-Churn) as shown by the darker line, assuming we wanted to use the model to correctly identify around 80% of the churners, then the chart indicates that around 25% of the non-churners would be incorrectly classified as people who churned. In other words, the ROC curve shows that if we increase our demand for accurately identifying the true positives, we in

turn incur an increasingly larger proportion of false positives.

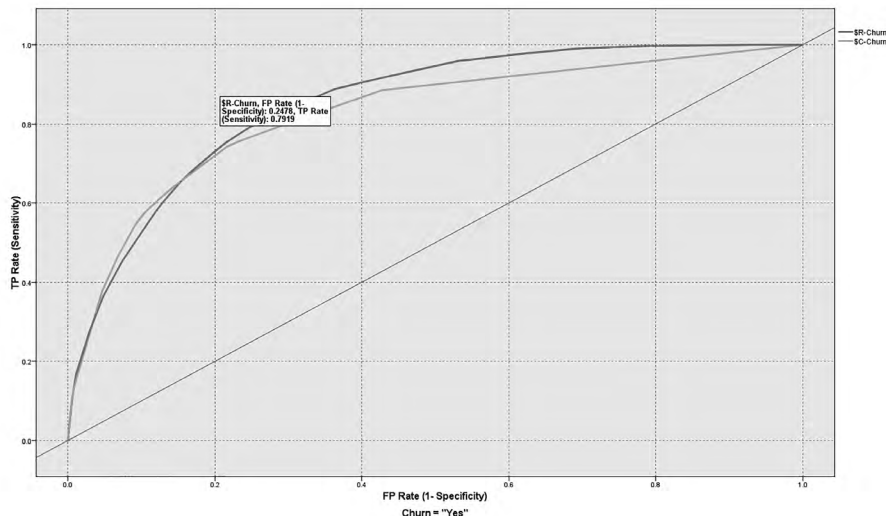


Figure 40 - ROC curve showing the model performance in terms of the relationship between true positives and false positives

It's worth noting that the examples we've looked at so far are normally applied to models with target categories. Even then, there are other kinds of charts that might be used in these situations such as lift charts, K-S charts and Lorenz curves. As there is no single way to represent the performance of a predictive model, most experienced analysts make sure that they are aware of at least a few different options.

6.8 Model performance metrics

Just as there are multiple ways in which analysts can illustrate model performance in charts, there are even more metrics available that attempt to summarise performance as a single number. These metrics are regularly used by sites such as [Kaggle.com](https://www.kaggle.com) that host competitions in order to choose a winning analytical model.

In Kaggle competitions analysts are invited to download a dataset with a view to taking part in challenges such as predicting survival on the Titanic, estimating property values or building a model to identify images of dogs vs. cats. With predictive modelling challenges, the competitors are provided with a sample dataset where the target outcome is shown for model training. Once a competitor has developed a potential predictive model, they can use it to generate predictions against a scoring dataset where the outcome is not known to them. They can then upload the scored dataset containing the predictions to the Kaggle site which in turn automatically calculates how well the model performed and how it ranked against all the other competitors' models.

Of course, in order to do this the competition organisers have to choose a specific criterion (or metric) against which to judge the performance accuracy of each entry. Here are just a few of the metrics that analysts and sites like Kaggle use to compare the performance accuracy of predictive models. For simplicity, I'll restrict the examples to those that may be used with categorical target variables.

6.8.1 Overall accuracy

This is simply the overall (or average) accuracy the model exhibits when considering all the groups in the target field. The weakness with this measure is that if the target contains group sizes that are very different, the model may appear to be more accurate than it deserves by predicting every record to belong to the same category. In situations like outbound marketing campaigns, the data may show that only 1% of people respond to an offer so using overall accuracy means that the model can appear to be 99% accurate by predicting everyone to be a non-responder. For this reason, overall accuracy may be a more appropriate measure when the target category proportions are closer to 50/50.

6.8.2 Area under the curve

The area under the curve (or AUC) measure is related to the ROC charts that we explored earlier. AUC is based on the principle that the larger the area under the ROC curve the better the model. Recall that the gains chart displays a diagonal line representing a random classifier. The diagonal line simply indicates that if you were using a random model you would expect to find 50% of the responders (or churners) from randomly sampling 50% of the data.

The AUC score for a model that followed the random line would therefore be 0.5 (scores lower than 0.5 would indicate the model was worse than random). A score for a perfect classification model would be 1. In practice this means that the range between a random model and perfect model is from 0.5 to 1. It depends a bit on the context but in most situations an AUC value of 0.8 or higher would be regarded as pretty good.

6.8.3 Gini coefficient

Gini coefficients are commonly used in credit scoring applications to choose models that will accurately predict the risk that a borrower will default on a loan. It's actually based on the area under the curve (AUC) measure that we just considered. For the record, the Gini coefficient is calculated as $2 * AUC - 1$. The Gini measure normalises the AUC metric in such a way that the values for random models and perfect models run from 0 to 1 respectively which some regard as a more intuitive range.

6.8.4 Lift

The lift value is a measure of how much better the model does at predicting the outcome compared to a random approach. If 10% of a dataset is comprised of

customers who have churned, a random model predicting every case to be a churner would be right about 10% of the time. This would generate a lift value equal to 1.0. If, however, the model was better than random and was able to predict the proportion of churners with 20% accuracy, it would be twice as accurate as the random approach and would generate a lift value of 2.0. The higher the lift value, the better than random the model is. Often lift measures are calculated on a proportion of the data, say the top 30%, where the model is most confident.

6.9 Model validation

Earlier we looked at the methods that model-building algorithms such as neural networks employ to prevent creating overfitted models. Remember that an overfitted model is one which is overly influenced by the idiosyncrasies and random fluctuations of the sample it was trained against. This means that it may appear to be accurate, but when it's applied to new data, the accuracy is greatly degraded.

Ideally, we want the model to correctly encode the relationships between the predictor (input) fields and the target field that exist in the wider population rather than the sample dataset it just happened to be built with. That's why getting an unbiased and representative sample is so important for any analysis we do.

To minimise overfitting, algorithms like neural networks hold back a randomly allocated portion of the data from the model-building process (e.g. 30%). This way the algorithm never gets to see all the data and so the model is effectively built against a sample of a sample. Withholding some data from the model-building process is generally a good idea for any attempt to develop a predictive model. Doing so will help us to ascertain how well the model is likely to perform when deployed against data that it hasn't encountered before, such as when we use it to make decisions in the real world. By holding some of our sample data back, we can validate the model. With that in mind, it makes sense for us to look at the different methods analysts use to validate their models.

6.9.1 Training / testing sample split

One simple way to validate a model is to split the data into two samples. Depending on the software used, this may require you to create two physically separate data sources or to at least filter some of the data out. This is normally done on a random basis. Some analytical platforms allow the user to create a randomly generated field that marks each case as 'training' or 'testing'. The model-building algorithm then ignores the testing cases and uses only those marked 'training' to build the model.

In terms of the proportional split between these two groups, it's entirely up to the analyst but generally speaking, one aims to have a large enough sample to build the model against (the training group) but still leave enough cases to validate the model performance (the testing group). The analyst could opt for a simple 50/50 split, although a split of 70% for training the model and 30% withheld for testing purposes is more common (this is due to the paucity of data that analysts often had

to work with historically). Using a training/testing regime means that the analyst pays particular attention to model performance on the testing sample when trying to decide which model to choose.

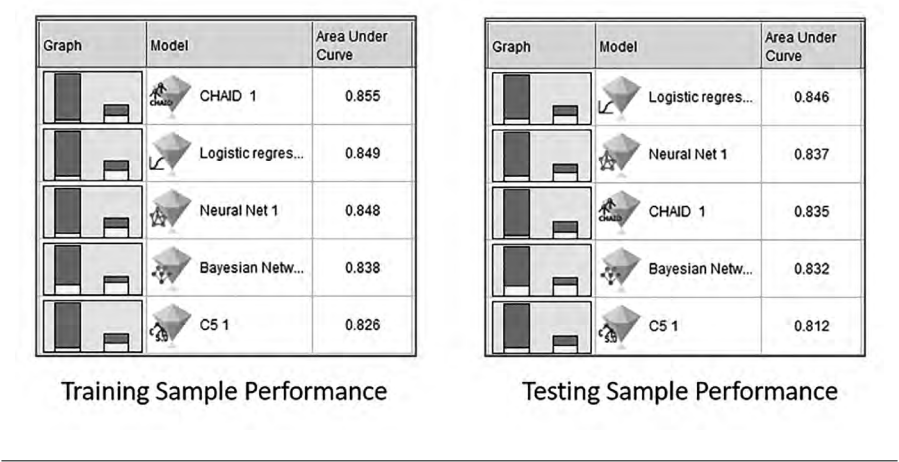


Figure 41 - Comparison of model performance between training and testing samples (test group based on a random 30% sample of cases)

As Figure 41 above illustrates, the best performing model developed against a training sample may not be the best performing model when applied to a test sample. The figure shows five predictive models built using the IBM SPSS Modeler Auto Classifier procedure. This procedure automatically tries out a number of algorithms and chooses the highest-ranking ones based on a pre-specified performance criterion.

In this case, the criterion was the area under the curve (AUC) metric. You can see from the image that in the training sample the best performing model was a CHAID decision tree, but when applied to the testing sample this particular model slipped to third place. The model with the highest AUC value in the testing sample was the logistic regression model. It's worth mentioning here that some analysts feel that relying purely on test datasets to validate a model isn't enough to be sure that the model you eventually select is the best one. So sometimes a third sample, known as the validation set, is defined to check the model performance.

6.9.2 Cross-validation

One of the issues with using a training/test sample split for validation is that it assumes you will have sufficient data to create two large enough random groups in order to build an effective model and adequately test it, but what if that's not the case? Approaches like cross-validation may offer a solution.

A popular form of this this method is the K-Fold cross-validation procedure. This technique works by randomly splitting the data into K equal-sized subsets for testing model performance (see Figure 42). Often the default value for K is 10, so the

procedure effectively creates 10 ‘folds’ (i.e. 10 separate sub-samples).

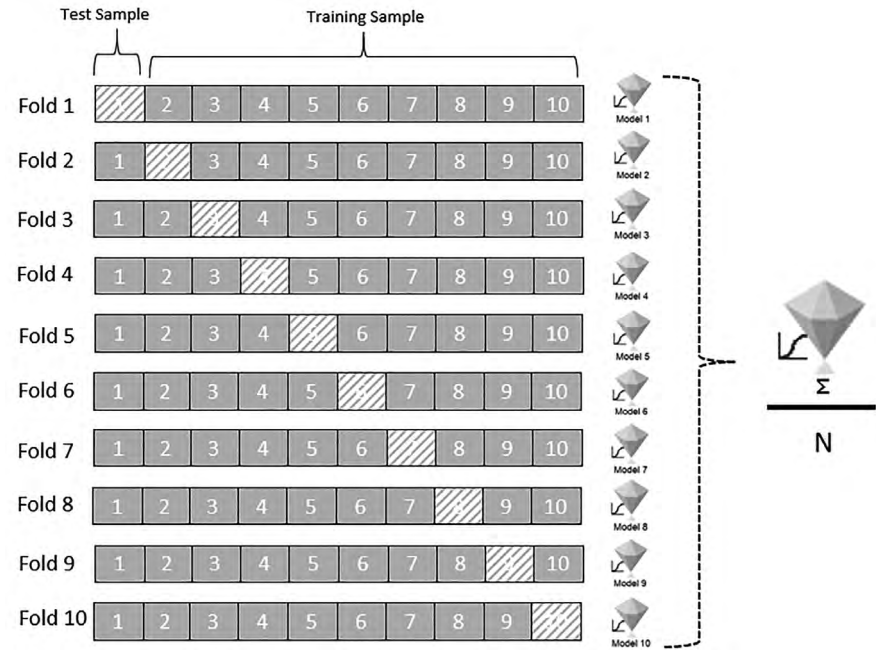


Figure 42 - Illustration of 10-Fold cross-validation procedure

The routine begins by withholding the first of the 10 folds and building a model on the remaining data in the sample. It then tests the model performance against the withheld data in the first fold. Having completed this first iteration, it releases the data from the first fold and withholds the data in the second fold before building a separate model on the remaining data and testing that model’s performance on the second fold. It continues to do this, working through each fold, building and testing each model at each step until 10 separate models have been built and tested on the 10 separate sub-samples. The results are then averaged to create an overall estimate of performance.

The obvious disadvantage with cross-validation methods is that they require a model to be trained repeatedly so are significantly more resource intensive. In fact, it’s possible to use the leave-one-out cross-validation method (also called the jack knife method) mentioned earlier. This is a more exhaustive version of cross-validation as the model is systematically trained on all the cases except one single observation. At each stage the model is then tested against the individual set-aside case before withholding the next case and repeating the process for every record in the dataset. The performance results from every iteration are again averaged together to compute a final overall score.

It is important to remember that accuracy is not the only criterion for choosing a final model. In an absolutely ideal world, most analysts (and organisations) would prefer to work with predictive models that are highly accurate, simple, transparent, insightful, stable over changing conditions, scalable, easy to deploy, and generating truly useful values in a sufficiently timely manner that they can be acted upon.

That's no easy ask and it highlights why the development and operational deployment of models should not be the sole responsibility of analysts. There are so many factors that need to be considered, especially when they have an impact on the wider business, that often it's a balancing act between the various pros and cons of one model over another. For that reason alone, it makes sense that the different stakeholders should work as a team to help make those choices. Ultimately, predictive analytics is about driving more informed decision-making, and in the next chapter we will explore what happens when a model is deployed in the real world.

CHAPTER 7

BACK IN THE REAL WORLD

You may recall that at the very start of this book, I wondered about the absence of stories concerning predictive analytics failures. Of course, when it comes to innovative applications, unless these failures are particularly spectacular or high profile, they aren't likely to be very newsworthy so it's little wonder that they don't garner much interest. Nevertheless, in my own experience a very large proportion of advanced analytics projects fail to make a noticeable impact in the real world.

The next time you meet a data scientist or predictive analytics expert, ask them which application they've worked on made the biggest difference and how they measured its effects? Don't be surprised if they have to really think about it for a minute. I'm not suggesting that there aren't many benefits to working through the CRISP-DM process even if the results are never deployed. For instance, just addressing the business understanding phase, or assessing the quality of the data, or figuring out what would be required to achieve the end goal can be extremely valuable, not to mention the useful insights that data exploration and modelling might uncover.

However, predictive analytics applications create data with the intention that this new information can be acted on. In the CRISP-DM process, this is what is meant by deployment - that the values a predictive model generates will be used to actively target opportunities or mitigate threats; the recommendations from an association model will be used to present customers with new offers or employees with suggested actions; anomaly detection models will alert security systems to unusual activity; and segmentation models will present clients with customised content when visiting websites.

It's precisely because deployment implies changing organisational behaviour, or at the very least, testing out how the new predictive analytics application performs in the real world, that this phase of CRISP-DM is often the most hazardous in terms of the project's success. If we don't anticipate and plan for deployment effectively, it's likely that the main output of the initiative will just be a report. And whilst reports have the potential to change behaviour and improve decision making, they also have the potential to be ignored.

There are lots of reasons why deployment often proves to be so difficult for organisations. One of the stumbling blocks often encountered in this phase is that model deployment can encroach on the activities of other roles within the organisation. For example, if the application's deployment requires that it makes regular updates to a customer database or website, the IT department may need to be involved in overseeing and validating this process. Alternatively, the project might be focussed on identifying customers for inclusion in a marketing campaign, so key personnel and processes within the marketing department will be affected. Whether a model is used to generate real time recommendations in a call centre environment

or create maintenance schedules to inspect ‘at risk’ assets, it’s going to affect someone else’s job. Of course, if the project team have adhered to a methodology like CRISP-DM, this is something they will have anticipated and planned for in the business understanding phase. So again, by the time the analysts reach this phase, they should already know how the deployment will be executed, who will be impacted and what constitutes success.

In this chapter we will look at the various ways in which deployment can occur. In most situations this will take the form of generating specially selected lists or data files with values that can be acted upon. This information can then be exported for use by an operational system such as a campaign management platform, a work roster, a maintenance scheduler or even an app on a customer’s phone.

However, we may also need to provide evidence that these new data actually help to improve the outcome in question, so we should think about the ways in which we can test and measure the impact of the application and how we might surface that information through a BI/MI tool for management. Figure 43 shows the deployment phase broken down into a series of tasks and sub-tasks. As with all stages of CRISP-DM, there is a strong emphasis on documenting every aspect of this project phase.

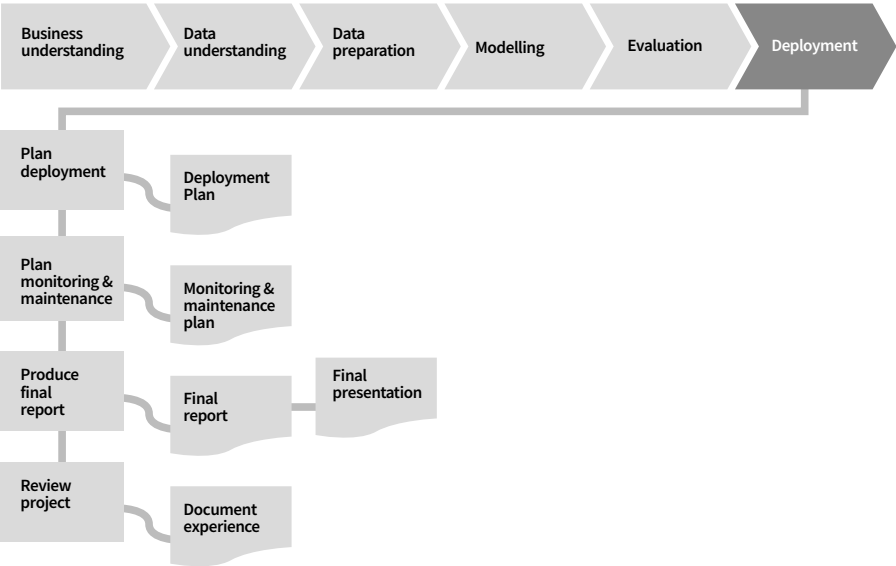


Figure 43 Breakdown of the deployment phase into tasks and sub-tasks

7.1 Creating selections

As I mentioned earlier, the process of applying a model to a dataset to generate new values (e.g. predictions) is referred to as scoring the data. That means we can use a

model that was developed to predict churn on historical data to score a new dataset containing information about current customers.

The effect of this is to create new fields that show the likelihood of each current customer cancelling their subscription or contract. Obviously, we do this in order to identify customers who have a higher than average likelihood of defecting within a given time period so that we can take some sort of action to retain them. That action may have costs associated with it, so it makes sense to think about how many current customers we can afford to target.

Perhaps the current average churn rate is around 5% per quarter. Technically, anyone with a churn likelihood above this number has a higher than average chance of leaving. Even if they are twice as likely to churn and so have a likelihood value equal to 10%, there's still a 90% chance that they will remain a customer. Immediately we have to address the very practical problem of how certain we want to be in our predictions and how many customers we can afford to target. Having made those decisions, we can then generate a list containing a specific number of customer IDs whose likelihood to churn is above a pre-determined threshold.

Figure 44 below shows a partial view of a data file that has been scored using a churn prediction model. Note that the field 'Churn_Prediction' is comprised simply of the categories 'Yes' and 'No' to indicate whether the model thinks each customer will churn. These values are based on whether or not the field 'Probability_of_Churn' is above or below the 0.5 (50%) probability threshold. The first case with customer ID 6867 is predicted to be a churner as the probability of churning is 0.607 (60.7%) whereas the second case with customer ID 4168 is predicted to remain a customer as the associated churn probability is only 0.058 (5.8%).

CustomerID	Churn_Prediction	Probability_of_Churn
6867	Yes	0.607
4168	No	0.058
3474	No	0.063
1934	No	0.060
764	Yes	0.601
2127	Yes	0.572
5828	Yes	0.517
1025	No	0.014
998	Yes	0.519
5250	No	0.008
6246	No	0.073
5687	Yes	0.786
2698	No	0.068

Figure 44 - A scored data file containing Customer ID values and model prediction fields that indicate each customer's likelihood of churning

We could of course simply select everyone that the model categorises as a churner – all those in the ‘Yes’ group. For each selected customer their likelihood of churning is greater than not churning. But we’re only seeing a partial view of the scored data file here, and therefore we can’t tell how many people the model has predicted will churn. In fact, the full data file contained 60,000 current customers. If the model only predicted a thousand of them to be churners and we have capacity to target more than that, maybe we should also include customers whose likelihood of defecting is less than 50%? To understand the variation in the model’s predictions we can visualise this distribution by using a simple histogram. Figure 45 below illustrates this.

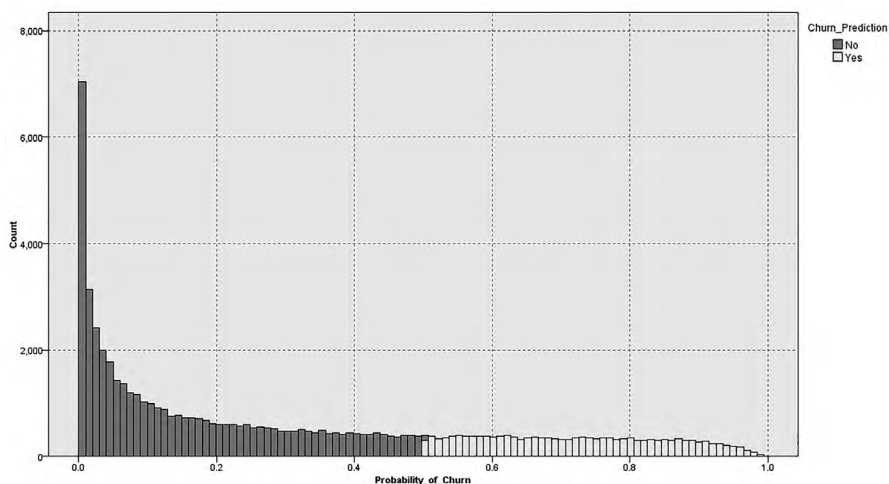


Figure 45 - Histogram of churn probability scores generated by a model

Histograms are designed to help us visualise a distribution by slicing it into an arbitrary but equal range of values. Each range is shown as individual bars or so-called bins. In this example, the histogram is quite fine-grained as there are around 100 bins in total, each with a range of about 0.01 (1%).

The first thing we notice is that the model has more cases with a low probability of churning than with a high probability of churning. That’s not unusual, and in fact I’ve worked with many predictive models where the probability of the event in question never even gets above 30%. In this example, the model calculated that around 26% of the customers (15,477) have a probability of churning above 0.5. If, however, our work during the business understanding phase indicated that we had enough budget to contact the 20,000 customers with the highest risk of churning, then we could select those customers whose probability of churn was around 0.4 (40%) or higher. Figure 46 illustrates this threshold in the histogram.

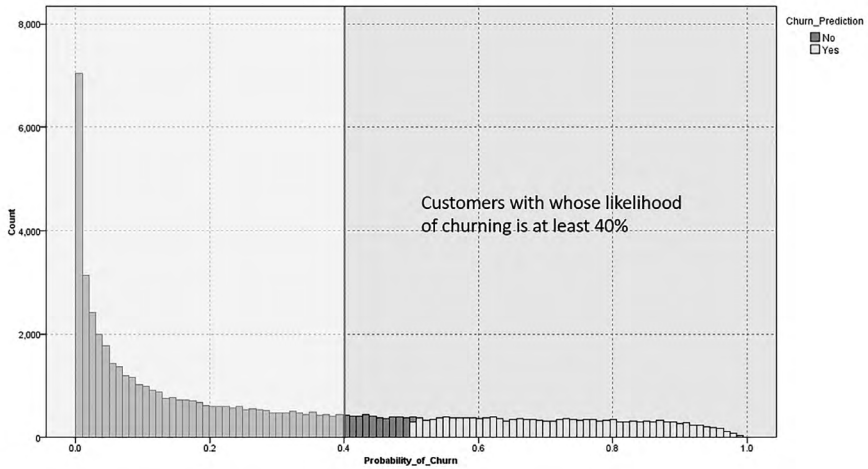


Figure 46 - Selecting circa 20,000 customers with the highest risk of churn

Figure 47 shows a sample of these selected customers. As you can see, although some of the customers are predicted to not churn, each record has a churn probability of at least 0.4. From a technical perspective, the physical deployment of this information could be as simple as exporting the data as a text file that just shows the customer ID values or inserting a table with the same information into an existing database. These data can then be used on an operational basis for selecting current customer targets as part of a proactive retention campaign. Likewise, we could use the gain charts from the previous chapter to select cases based on a threshold, where for example, we might want to generate a selection that will capture around 60% of all the expected churners.

CustomerID	Churn_Prediction	Probability_of_Churn
335.000	No	0.402
6814.000	Yes	0.513
3801.000	Yes	0.607
5653.000	Yes	0.538
784.000	Yes	0.608
6778.000	Yes	0.860
5862.000	Yes	0.928
2971.000	Yes	0.928
5268.000	Yes	0.695
6856.000	No	0.465
5912.000	Yes	0.580
1524.000	Yes	0.779
801.000	Yes	0.888
5756.000	Yes	0.692
1494.000	Yes	0.793
6026.000	Yes	0.846
2031.000	Yes	0.909
692.000	No	0.476
2195.000	Yes	0.620
1401.000	Yes	0.840

Figure 47 - Sample list of customers with a churn risk of at least 0.4 (40%)

Simply exporting a list of addresses or ID values is one of the most straightforward ways in which model scores can be deployed. It's worth noting that in some situations the model that generates these scores can itself be directly incorporated into third party systems such as campaign management tools. This is usually done by exporting the model in a common interchange format such as PMML (predictive model markup language). PMML is an XML-based coding standard that acts as a lingua franca between different systems so that, for example, a decision tree model created by one platform can be correctly understood and utilised by another.

Another way in which scored cases can be selected is that, rather than including them based on a probability threshold, we can take into account the costs and potential benefits associated with choosing them as deployment targets. In these situations, we're directly taking note of the costs associated with false positives and false negatives.

Let's imagine that the business understanding phase has determined the goal of the application should be to make our customer retention campaign profitable. The context here is that costly retention offers are being made to customers with a high risk of churning. These offers take the form of extending their contracts by two months but with free telephony charges.

Unfortunately, many of those who take up the offer churn after the two months anyway leading to a net loss of £140K in each campaign. Ideally, the company would like to ensure that each campaign makes a net profit of £70K. Let's assume that the

organisation has calculated that those who defect after accepting the offer cost the company around £20 each and that each person who remains a customer, generates around £62 in revenue. In the previous, chapter we saw how gains charts can be used to show us how many records we need to include in order to gain a certain percentage of a target group. Profit charts, such as the one shown in Figure 48 can be used to find the optimal selection of cases for inclusion in a campaign with a view to making the campaign as cost effective as possible.

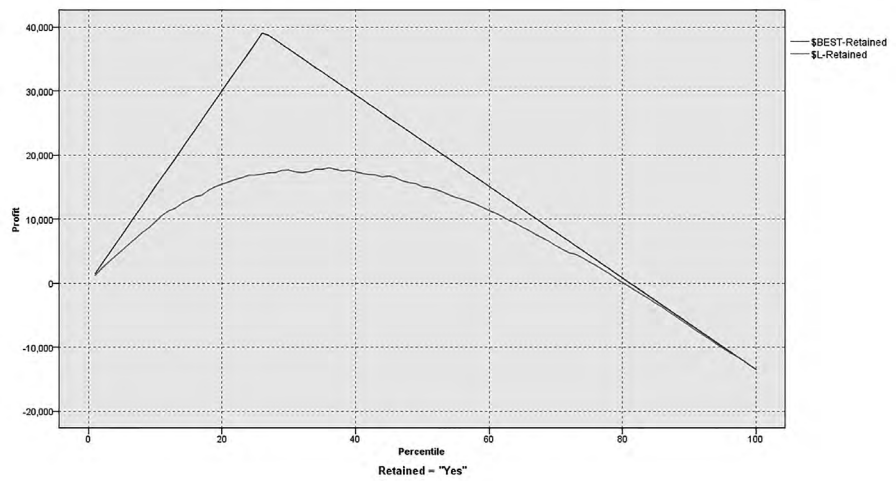


Figure 48 - Profit chart showing estimated campaign profitability based on expected revenue of £62 for successfully retaining a customer and expected offer costs of £20

Although the graph looks a little different from the gains charts that we have seen before, like those charts, it's based on the model training data because we need to know the actual outcome to calculate the chart's values.

The easiest way to interpret the profit chart is to first look at the apex of the best line near the 26th percentile on the horizontal axis. This top line, labelled \$BEST-Retained in the legend, indicates that if a model was able to successfully identify all those customers that the offer helped retain, with no one churning after offer period ended, then the campaign would generate about £40K profit (assuming the campaign size was equal to the sample size in this example). Conversely, the lowest point of the line (near the 100th percentile) indicates that if we were to make the offer to everyone in this sample dataset, irrespective of their likelihood to be retained, then this would generate a loss of around £13.5K.

The curved line (labelled \$L-Retained) represents the actual predictive model. We can see that the apex of this line is around the 36th percentile (see Figure 49). If the model is reliable this means that selecting the 36% of cases with the highest probability of being retained should generate the highest estimated overall profit for the retention campaign, which in this case would be £17,990.

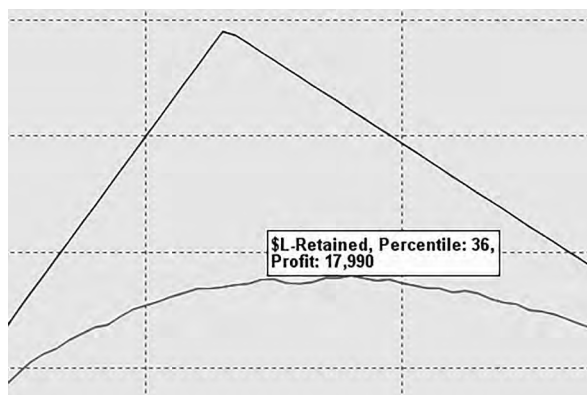


Figure 49 - Selecting the 36% of customers with the highest probability of being retained generates the largest estimated overall profit for the retention campaign

In fact, IBM SPSS Modeler allows us to directly generate the chart's selection formula to use when we deploy the model in order to score the much larger file of current customers. This means that, given the volume of cases in the current customer file, the desired threshold of £70K profit should easily be reached.

It's also worth pointing out that the revenue and cost values don't have to be fixed numbers. They may vary from one customer to another, so we could instead base our calculations on actual fields that represent revenue and costs as variables. Lastly, although this example is based on a marketing campaign, we could just as easily use this kind of chart to make selections based on factors such as customer satisfaction values, system downtime or asset repair costs. Again, this kind of deployment strategy implies that the project's stakeholders had already anticipated how they would use a resultant model long before they actually got to the deployment phase.

7.2 Testing deployment

Of course, the ultimate test of our application is whether or not it actually works. We have now worked our way through the entire process of planning a project, auditing the available data, preparing it and building models, before finally assessing their performance and deploying the results. But we will still need to show that our efforts deliver sufficient improvements to be regarded as a valuable, or at least a worthwhile endeavour. I often find it surprising as to how little thought is given to this critical aspect of the initiative. The credibility of the entire project, and the likelihood that the organisation will support similar projects in the future may rest on how well we measure and prove the benefits of the application in question.

If the project team have been sufficiently thorough during the business understanding phase, they will already have measured how well the organisation's current efforts, in the absence of a predictive analytics application, address the problem in hand.

In other words, it's important to establish a baseline that shows what happens if a useful model isn't used.

To measure the model performance in the real world some sort of testing regime will need to be designed. In situations where the deployment phase entails scoring current data to take some sort of proactive action, such as contacting customers to dissuade them from churning, inspecting machinery that are likely to require maintenance, presenting customised content on mobile devices or investigating transactions that appear suspicious, it will probably be necessary that this action isn't taken in every situation or case that the model identifies as relevant. This kind of approach is analogous to a clinical trial where the use of a placebo or a standard treatment regimen is compared against the outcomes recorded using a new experimental drug.

Following on from our earlier illustration of predicting customer churn, let's imagine that the company in question has decided to test the selected model against customers who are approaching their contract end dates. It's essential that whatever the offer is, it's sufficiently compelling to be attractive to a wide range of customers. Let's assume that the offer takes the form of providing an additional service at no extra charge if the customer renews their contract.

With that in mind, the project team create three groups of current customers: group A contains a random selection of customers that will receive no proactive retention offers to dissuade them from churning; group B contains a separate random selection of customers who all receive proactive retention offers irrespective of their likelihood to cancel their contracts; lastly group C also contains a random list of customers. However, in group C only those customers that the predictive churn model estimates as having a high likelihood of churning will receive the proactive offer. Figure 50 illustrates this scenario.



Figure 50 - Creating test groups to establish model performance in the real world

We should always bear in mind that our model was trained against historical information and that the current customer data may reflect different market conditions, so we shouldn't be surprised if it doesn't perform quite as well as it did during the evaluation phase. Nevertheless, if the model correctly identifies those at risk of churning and the offer is sufficiently compelling, we should at least be able to demonstrate that it is better than random at reducing customer churn.

In our example, we would wait until all the customers' contracts in the three samples had passed their renewal dates and then calculate what proportion of customers churned in each group. Having done so, we could carry out statistical tests to see if the differences in the churn rates were sufficiently large to be regarded as statistically significant.

It's reasonable to expect that group A would have the highest churn rate as this represents what happens when no offer is made to anyone. This may be close to the average churn rate metric that the project team hopefully established during the business understanding phase. Group B will probably have the smallest churn rate as the offer is being made to everyone in a random selection containing both customers that are unlikely to defect as well as those that are at high risk of doing so. Group C should have a lower churn rate than group A but probably a higher churn rate than group B, because although not all the customers in this group were incentivised to renew their contracts, those that the model estimated to be relatively high risk did receive the proactive offer.

Comparing the churn rates between group A and group B will help the team to figure out how compelling the offer is at preventing churn for all customers. Comparing the churn rate for the selectively targeted list in group C to those in group A where no offer was made, should help establish how well the model performs against a random baseline and therefore how valuable it is.

Of course, there are lots of other analyses that we could perform here to test the model. If we scored all the customers beforehand then we could look at those people in group A with a high risk of churning and who received no offer, and compare that churn rate to those customers in group C with a similarly high churn risk but who did receive the offer. That way, we could establish how well the offer dissuaded potential churners from cancelling their contracts.

Alternatively, we could look at those with a low risk of churning who received the offer and who churned anyway, compared to those with a similarly low risk who didn't receive the offer and who also churned. This comparison might help us to establish if the offer actually stimulated people to cancel their contracts by reminding them that their contract renewal date was approaching.

In fact, we could go even further and randomise the kind of retention offers that are being made in the first place. This would allow us to build a separate model to help predict which offer type would have the highest chance of retaining different at-risk customers. The example illustrated here is by no means the definitive method by

which analysts can test model performance in the real world and establish its value, but hopefully it serves to illustrate the kind of empirical approach that project teams often employ to prove the worth of a predictive analytics application.

There is another reason why it is useful to create random control groups when working with deployed models. At some point in the future, we may need to create a new predictive model and apply it to that future period's current customer base. This is because the accuracy of any predictive model will eventually degrade. Perhaps due to changes in the market or its wider social context, the historical data the model was built upon may no longer accurately reflect the contemporary world.

The process of rebuilding an existing model by using more up-to-date data is known as a model refresh. If a previous churn risk model had been used to score the entire customer base in order to determine who receives a retention offer, then the very use of that model will have influenced the probability of churning for everyone in the data. Analysts often get around this problem by withholding a randomly chosen sample from the offer process to let those customers churn naturally. By doing so, they can use this uncontaminated sample to refresh the existing model at a future point.

Lastly, as Figure 51 reminds us, the outer circle in the CRISP-DM diagram denotes the fact that this entire process is iterative in nature. Predictive analytics applications can be improved and maintained through regular updates and additional efforts. What might have begun as a tentative pilot project to develop a basic model quickly can easily evolve into a much more mature and valuable operational concern requiring regular work cycles and acting as a template for other predictive analytics projects.

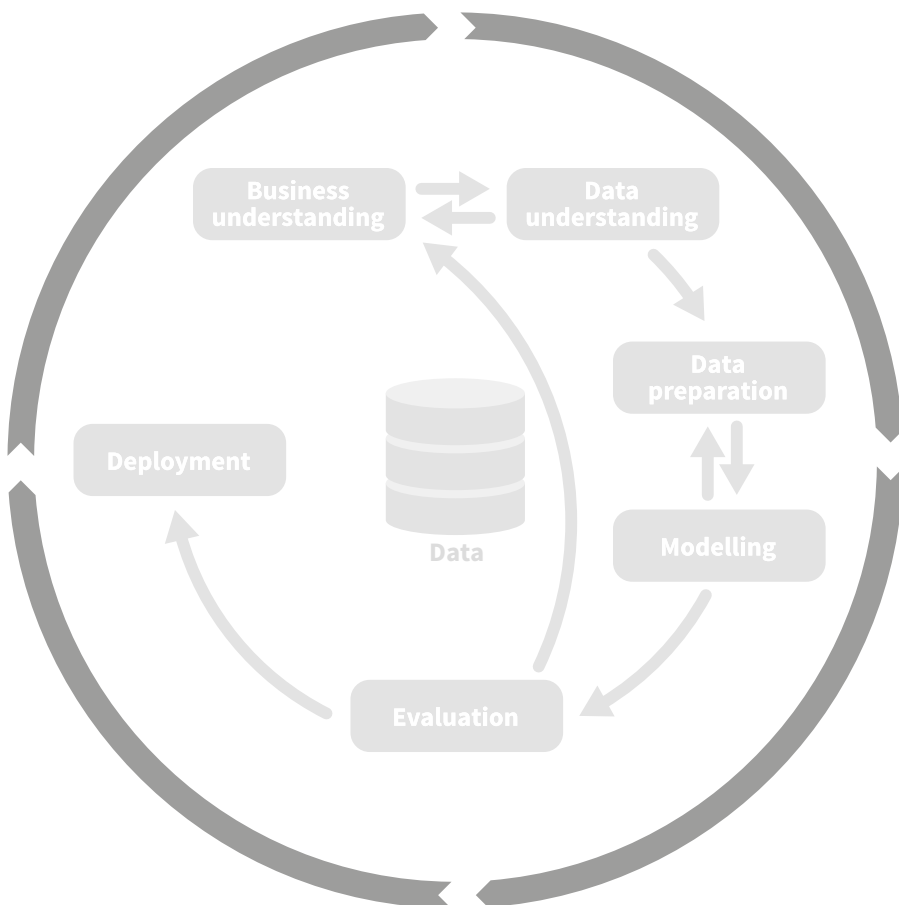


Figure 51 - The outer circle of the CRISP-DM diagram indicating the iterative nature of the process model

CHAPTER 8

BEYOND DEPLOYMENT

One of the criticisms of the CRISP-DM process model is that it doesn't have an explicit focus or phase for the kind of important tasks associated with previously deployed models. These tasks can include monitoring model performance, managing operational decisions and performing simulation analysis or scenario planning.

You may recall that the CRISP-DM methodology was originally conceived in 1996 and it's fair to say that quite a lot has happened in the world of advanced analytics since then. It's worth also noting that CRISP-DM is not the only recognised methodology available to analysts. Other examples include SAS Institute's **SEMMA**: an acronym that stands for sample, explore, modify, model and assess. Microsoft advocates its own **Team Data Science Process** (TDSP) and in 2015 IBM published a new methodology called the **Analytics Solution Unified Method** (ASUM). Needless to say, each of these approaches have their own merits, but the crucial point they make is that when it comes to initiatives that utilise advanced analytics in order to drive measurable improvements in decision making, it really helps to have a plan.

8.1 Monitoring performance

Many organisations that use predictive analytics as part of their day-to-day operations are keenly aware that model performance deteriorates over time. To that end, they often employ monitoring software displaying regular updates of model accuracy in the form of dashboard views. This is especially so when the companies concerned are working with multiple deployed models covering a range of targets.

The metrics and charts these systems display may be exactly the same as the ones we encountered in Chapter 6 when we explored how model performance is evaluated. Overall accuracy statistics like area under the curve (AUC) and visualizations such as ROC charts are used to constantly track the accuracy of the predictive analytics applications that underpin outbound marketing campaigns, online recommendation engines, fault detection systems and fraud management platforms.

Often these systems are developed on existing ML and BI platforms that are commonly used to measure KPIs (key performance indicators) and operational metrics. Sometimes they form part of a dedicated model management platform of the sort provided by companies like IBM, SAS Institute and Microsoft. These platforms are designed to allow the secure, centralised storage of analytical assets such as models and code and to enable them to be shared and managed by key stakeholders. They may also provide real-time deployment options so that model scores can be generated on-demand by users of websites, CRM systems and hand-held devices.

8.2 Automation

Doubtless most of these kinds of organisations will have already reached a

reasonably sophisticated standard in terms of their usage of modelling, whereas in this book we've really focussed on what it takes to ensure a successful outcome for an individual predictive analytics project. Still, it's not uncommon that, once a successful application has been developed and proven to have value, the project team are asked to look at ways in which these new capabilities can be scaled.

One obvious route is to investigate how the process might be automated. A simple form of automation may involve creating scheduled batch jobs that read the required data sources and prepare them before they are in turn automatically scored by a model with results being exported to a database or in a flat file format for actioning. More sophisticated forms of automation may be required if the same job has to be tightly coordinated with other key processes in the business.

In these situations, conditional execution may allow for specific steps in the data scoring process to be fired in sequence as new files are made available or when other procedures in the IT infrastructure such as back-ups have been completed. Often these kinds of automated processes are accompanied by clean-up steps if a procedure fails, with accompanying emails being dispatched to relevant personnel alerting them to newly generated reports or errors encountered.

One of the most sophisticated approaches to automation is the use of champion-challenger models. In these circumstances, the currently used model is labelled the champion and it remains in use until the system automatically replaces it with a new or refreshed model. To do this, each time the training data is updated one or more challenger models are automatically created.

Using either the same techniques or possibly including other model-building algorithms, the performance metrics of newly built challenger models are then compared to the current champion model. If a new challenger is deemed to have performed more effectively against the specified accuracy criterion, then it becomes the new champion model. By using champion-challenger regimes, organisations can ensure that they have the most accurate and up-to-date models in deployment without requiring manual intervention.

8.3 Planning and deciding

A further aspect of working with deployed models is the use of simulation or scenario planning. You may find that this kind of work is also sometimes referred to as 'what if' analysis. Put simply, it is used to uncover what effects changing the values of the model inputs have on the estimated outcomes. For instance, returning to our churn example, the project analysts might have incorporated information about their competitors' tariffs to help predict the likelihood that a customer will cancel their contract, the logic being that their more price-sensitive customers are likely to switch provider if a competitor is offering a cheaper tariff at the time their contract expires.

Using simulation techniques, the analysts could create a new artificial dataset where the competitor tariffs are allowed to vary across a pre-specified range of prices.

The analysts can then establish how many more customers are likely to churn in a market where a competitor decreases the price of a key tariff by 10%. Conversely, they can also see how many fewer customers will defect if the company reduces their own pricing by a specific amount. Scenario planning is therefore a useful way for businesses to estimate the effects of unforeseen changes and test the resilience of key systems by incorporating the sorts of models on which predictive analytics applications rely.

Lastly, we return to the ultimate goal of predictive analytics: making better decisions. Indeed, the very problem of trying to use analytics to drive more effective actions is often a key aspect of disciplines such as decision management. In this context, decision management encompasses the development and execution of systems designed to manage and control automated decision-making so that organisations may optimise their interactions with customers, employees and suppliers.

Decision management systems can use rules and model scores from predictive analytics to provide rapid, high-volume recommendations and estimates in real time. Decision management as a proposition becomes especially compelling when multiple analytical models are generating different values for separate aspects of an interaction. These aspects could include trying to estimate an individual's level of satisfaction, their appetite for accepting an additional service, the risk that they might cancel a transaction or their likelihood to commit fraud.

Such technologies may be used in handling insurance claims where the data gathered in conversation with claimants can be scored by various analytical models so that the decision management system can in turn recommend the most appropriate action to the claim handler. The resultant recommendations could be anything from prioritising payment through to alerting the claim investigation unit. In the same way, a field technician could inspect and upload diagnostic data from an aircraft engine that in turn triggers an anomaly detection model and causes the decision management system to recommend additional diagnostic steps.

To many organisations that have barely begun to investigate the capabilities of predictive analytics, these examples may seem impossibly exotic or unrealistic, but we should remember that they all began with someone trying to simply account for the variation in a key outcome and then thinking about how they could exploit this information to better effect.

8.4 Last thoughts

When I began my analytics career in the early 1990's, the projects my colleagues and I worked on, and the techniques we employed, meant that most people regarded us as statisticians. At some point we were deemed to be data miners, then we became predictive analytics consultants before seamlessly blending into data scientists. Now apparently, we all work in AI.

Since 2003 I've tended to stick with the predictive analytics badge, not because it's a particularly accurate characterisation of the field I work in, but because it's the least worst description of the ragbag of disciplines, traditions and tools we employ to help clients make better decisions with data.

Clearly, as we enter the third decade of this century, advanced analytics applications are very much in vogue. More people than ever are interested in finding out how they can make better use of their data resources. When we speak to attendees at events organised by Smart Vision Europe, a recurring theme is that they feel their organisations ought to be doing more in this area but they're not always sure where or how to begin. Generally, our advice is as follows:

- Begin with a small-scale project
- Focus on a topic which is measurable and valuable with variable outcomes
- Work collaboratively using a written methodology
- Think about how you would measure the usefulness or accuracy of the application's output and how it could drive different behaviours

Lastly, I think it's worth noting that after 30 years working in this area, I'm struck time and again by the fact that the most successful applications I've encountered rarely depended on the use of state-of-the-art algorithms or the talents of brilliant analytical minds. True innovation may always require something of a pioneering spirit, but this is one area where you should never underestimate what can be achieved by a small team of business-focussed, data-literate people with a plan.

Good luck.

CHAPTER 9

BIBLIOGRAPHY

- Anon., 2020. [Online]
Available at: https://en.wikipedia.org/wiki/Predictive_analytics
- Anon., 2020. *IBM.com*. [Online]
Available at: <https://www.ibm.com/uk-en/analytics/predictive-analytics>
- Breiman, L., 2001. Random Forests. *Machine Learning*, pp. 5-32.
- Breiman, L., 2001. Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3), pp. 199-215.
- Daniel, E., 2019. *Predictive analytics names Rugby World Cup winner*. [Online]
Available at: <https://www.verdict.co.uk/rugby-world-cup-results/>
- Diamond, J., 1997. *Guns, Germs and Steel: A short history of everybody for the last 13,000 years*. London: Vintage.
- Friedman, J. H., 1997. *Data mining and statistics: What's the connection?*. s.l.: s.n.
- Geological Society of America, 2019. [Online]
Available at: <https://phys.org/news/2019-09-machine-reveal-geology-humans.html>
- Gordon, R., 2019. *Using machine learning to estimate risk of cardiovascular death*. [Online]
Available at: <http://news.mit.edu/2019/using-machine-learning-estimate-risk-cardiovascular-death-0912>
- Goscha, M., 2019. *Ping An Technology AI predicts flu outbreaks with 90% accuracy*. [Online]
Available at: <https://technode.com/2019/09/24/ping-an-technology-ai-predicts-flu-outbreaks-with-90-accuracy/>
- Hilton, 2018. *Corporate report*, s.l.: s.n.
- HMRC, 2019. *2018-19 Annual Report and Accounts*, s.l.: s.n.
- Johnson, C., 2012. *Netflix Never Used Its \$1 Million Algorithm Due To Engineering Costs*. [Online]
Available at: <https://www.wired.com/2012/04/netflix-prize-costs/>
- Kass, G. V., 1980. An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics*, 29(2), pp. 119-127.
- Krill, P., 2001. Analytics redraw CRM lines. *InfoWorld*, 3 December, Issue 49, pp. 17-18.
- Leo Breiman, J. F. C. J. S. R. O., 1984. *Classification and Regression Trees*. s.l.: Chapman and Hall.
- Quinlan, R. J., 1986. Induction of Decision Trees. *Machine Learning*, 1(1), pp. 81-106.
- Sarle, W. S., 1994. *Neural Networks and Statistical Models*. s.l.: s.n.
- Shearer, C., 2000. The CRISP-DM model: the new blueprint for data mining. *Journal of Data Warehousing*, 5(13), pp. 13-22.
- Silver, N., 2013. *The Signal and the Noise: The Art and Science of Prediction*. London: Penguin.
- Spiegelhalter, D., 2019. *The Art of Statistics*. s.l.: Pelican.
- Thames Water, 2019. *Building a better future: Annual Report and Annual Performance Report 2018/19*, s.l.: s.n.
- Vodafone, 2019. *Vodafone Group Plc Annual Report*, s.l.: s.n.
- Wenz, J., 2016. *Marvin Minsky, the Man Who Built the First Artificial Brain, Dead at 88*. [Online]
Available at: <https://www.popularmechnics.com/technology/robots/news/a19131/marvin-minsky-obituary/>

The media is awash with stories heralding the rise of machine learning and AI. Everyone is talking about bleeding-edge applications that have outperformed humans and revolutionised business practices, but nobody's talking about all the projects that quietly ended in failure. Why did some of these initiatives succeed when others didn't? Very often, in the rush to embrace data science, businesses lose sight of the fact that real success needs to be planned for, and that algorithms are just tools that help to us to make better decisions. Drawing on decades of insider experience, Jarlath Quinn dismantles the jargon behind advanced analytics, reveals the historical context in which the data science industry developed and provides a roadmap for those curious about how Predictive Analytics can be used to successfully drive positive change.

Jarlath Quinn is a consultant for Smart Vision Europe. As a veteran of the analytical software industry, working under the auspices of SPSS, IBM and SAS Institute, he has spent over 30 years teaching and delivering solutions based on statistics and machine learning technology. He likes astronomy, fishing and pale ale.



Smart Vision Europe Limited, Burlingham House, Norwich Road, Saxlingham Nethergate, Norwich, NR15 1TP

www.sv-europe.com