DAVID SPIEGELHALTER

The Art of Statistics
Learning from Data

A PELICAN BOOK

PELICAN
an imprint of
PENGUIN BOOKS

PELICAN BOOKS

UK | USA | Canada | Ireland | Australia
India | New Zealand | South Africa

Penguin Books is part of the Penguin Random House group of companies whose addresses can be found at global.penguinrandomhouse.com.

Penguin Random House UK

First published 2019

007

Book design by Matthew Young
Set in 11/16.13 pt FreightText Pro
Typeset by Jouve (UK), Milton Keynes
Printed and bound in Great Britain by Clays Ltd, Elcograf S.p.A.

A CIP catalogue record for this book is available from the British Library

ISBN: 978-0-241-39863-0

Penguin Random House is committed to a sustainable future for our business, our readers and our planet. This book is made from Forest Stewardship Council® certified paper.

www.greenpenguin.co.uk

CHAPTER 6

Algorithms, Analytics and Prediction

The emphasis so far in this book has been on how statistical science can help us to understand the world, whether it is working out the potential harm of eating bacon sandwiches, or the relationship between the height of parents and offspring. This is essentially scientific research, to work out what is really going on and what, in the terms introduced in the last chapter, is just residual error to be treated as unavoidable variability that cannot be modelled.

But the basic ideas of statistical science still hold when we are trying to solve a practical rather than a scientific problem. The basic desire to find the signal in the noise is just as relevant when we just want a method that will help in a particular decision faced in our daily lives. The theme behind this chapter is that such practical problems can be tackled by using past data to produce an algorithm, a mechanistic formula that will automatically produce an answer for each new case that comes along with either no, or minimal, additional human intervention: essentially, this is 'technology' rather than science.

There are two broad tasks for such an algorithm:

- Classification (also known as discrimination or **supervised learning**): to say what kind of situation we're

facing. For example, the likes and dislikes of an online customer, or whether that object in a robot's vision is a child or a dog.

- Prediction: to tell us what is going to happen. For example, what the weather will be next week, what a stock price might do tomorrow, what products that customer might buy, or whether that child is going to run out in front of our self-driving car.

Although these tasks differ in whether they are concerned with the present or the future, they both have the same underlying nature: to take a set of observations relevant to a current situation, and map them to a relevant conclusion. This process has been termed **predictive analytics**, but we are verging into the territory of **artificial intelligence (AI)**, in which algorithms embodied in machines are used either to carry out tasks that would normally require human involvement, or to provide expert-level advice to humans.

'Narrow' AI refers to systems that can carry out closely prescribed tasks, and there have been some extraordinarily successful examples based on machine learning, which involves developing algorithms through statistical analysis of large sets of historical examples. Notable successes include speech recognition systems built into phones, tablets and computers; programs such as Google Translate which know little grammar but have learned to translate text from an immense published archive; and computer vision software that uses past images to 'learn' to identify, say, faces in photographs or other cars in the view of self-driving vehicles. There has also been spectacular progress in systems playing games,

such as the DeepMind software learning the rules of computer games and becoming an expert player, beating world-champions at chess and Go, while IBM's Watson has beaten competing humans in general knowledge quizzes. These systems did not begin by trying to encode human expertise and knowledge. They started with a vast number of examples, and learned through trial and error rather like a naïve child, even by playing themselves at games.

But again we should emphasize that these are technological systems that use past data to answer immediate practical questions, rather than scientific systems that seek to understand how the world works: they are to be judged solely on how well they carry out the limited task at hand, and, although the form of the learned algorithms may provide some insights, they are not expected to have imagination or have super-human skills in everyday life. This would require 'general' AI, which is both beyond the content of this book and, at least at present, beyond the capacity of machines.

Ever since formulae for calculating insurance and annuities were developed by Edmund Halley in the 1690s, statistical science has been concerned with producing algorithms to help in human decisions. The modern development of data science continues that tradition, but what has changed in recent years is the scale of the data being collected and the imaginative products being developed: so-called 'big data'.

Data can be 'big' in two different ways. First, in the number of examples in the database, which may be individual people but could be stars in the sky, schools, car rides or social media posts. The number of examples is often given the label n, and in

my early days n was 'big' if it was anything more than 100, but now there may be data on many millions or billions of cases.

The other way that data can be 'big' is by measuring many characteristics, or features, on each example. This quantity is often known as p, perhaps denoting parameters. Thinking again back to my statistical youth, p used to be generally less than 10 – perhaps we knew a few items of an individual's medical history. But then we started having access to millions of that person's genes, and genomics became a small n, large p problem, where there was a huge amount of information about a relatively small number of cases.

And now we have entered the era of large n, large p problems, in which there are vast numbers of cases, each of which may be very complex – think of the algorithms that are analysing all the posts, likes and dislikes of each of the billions of Facebook subscribers to decide what sort of adverts and news to feed them.

These are exciting new challenges which have brought waves of new people into data science. But, to refer yet again to the warning at the start of this book, these trough-loads of data do not speak for themselves. They need to be handled with care and skill if we are to avoid the many potential pitfalls of using algorithms naïvely. We shall see some classic disasters in this chapter, but first we need to consider the fundamental problem of boiling the data down into something useful.

Finding Patterns

One strategy for dealing with an excessive number of cases is to identify groups that are similar, a process known as

clustering or **unsupervised learning**, since we have to learn about these groups and are not told in advance that they exist. Finding these fairly homogeneous clusters can be an end in itself, for example by identifying groups of people with similar likes and dislikes, which then can be characterized, given a label, and algorithms built for classifying future cases. The clusters that have been identified can then be fed appropriate film recommendations, advertisements, or political propaganda, depending on the motivation of the people building the algorithm.

Before getting on with constructing an algorithm for classification or prediction, we may also have to reduce the raw data on each case to a manageable dimension due to excessively large p, that is too many features being measured on each case. This process is known as **feature engineering**. Just think of the number of measures that could be made on a human face, which may need to be reduced to a limited number of important features that can be used by facial recognition software to match a photograph to a database. Measures that lack value for prediction or classification may be identified by data visualization or regression methods and then discarded, or the numbers of features may be reduced by forming composite measures that encapsulate most of the information.

Recent developments in extremely complex models, such as those labelled as **deep learning**, suggest that this initial stage of data reduction may not be necessary and the total raw data can be processed in a single algorithm.

Classification and Prediction

A bewildering range of alternative methods are now readily available for building classification and prediction algorithms. Researchers used to promote methods which came from their own professional backgrounds: for example statisticians preferred regression models, while computer scientists preferred rule-based logic or 'neural networks' which were alternative ways to try and mimic human cognition. Implementation of any of these methods required specialized skills and software, but now convenient programs allow a menu-driven choice of technique, and so encourage a less partisan approach where performance is more important than modelling philosophy.

As soon as the practical performance of algorithms started to be measured and compared, people inevitably got competitive, and now there are data science contests hosted by platforms such as Kaggle.com. A commercial or academic organization provides a data set for competitors to download: challenges have included detecting whales from sound recordings, accounting for dark matter in astronomical data, and predicting hospital admissions. In each case competitors are provided with a training set of data on which to build their algorithm, and a test set that will decide their performance. A particularly popular competition, with thousands of competing teams, is to produce an algorithm for the following challenge.

> Can we predict which passengers survived the sinking of the *Titanic*?

On its maiden voyage, the *Titanic* hit an iceberg and slowly sank on the night of 14/15 April 1912. Only around 700 of more than 2,200 passengers and crew on board got on to lifeboats and survived, and subsequent studies and fictional accounts have focused on the fact that your chances of getting on to a lifeboat and surviving crucially depended on what class of ticket you had.

An algorithm that predicts survival may at first seem an odd choice of Problem within the standard PPDAC cycle, since the situation is hardly likely to arise again, and so is not going to have any future value. But a specific individual provided me with some motivation. In 1912 Francis William Somerton left Ilfracombe in north Devon, close to where I was born and brought up, to go to the US to make his fortune. He left his wife and young daughter behind, and bought a third-class ticket costing £8 1s. for the brand-new *Titanic*. He never made it to New York – his memorial is in Ilfracombe churchyard (Figure 6.1). An accurate predictive algorithm will be able to tell us whether Francis Somerton was unlucky not to survive, or whether his chances were in fact slim.

The Plan is to amass available data and try a range of different techniques for producing algorithms that predict who survived – this could be considered more of a classification than a prediction problem, since the events have already happened. The Data comprise publicly available information on 1,309 passengers on the *Titanic*: potential predictor variables include their full name, title, gender, age, class of travel (first, second, third), how much they paid for their ticket, whether they were part of a family, where they boarded the boat (Southampton, Cherbourg, Queenstown), and limited data

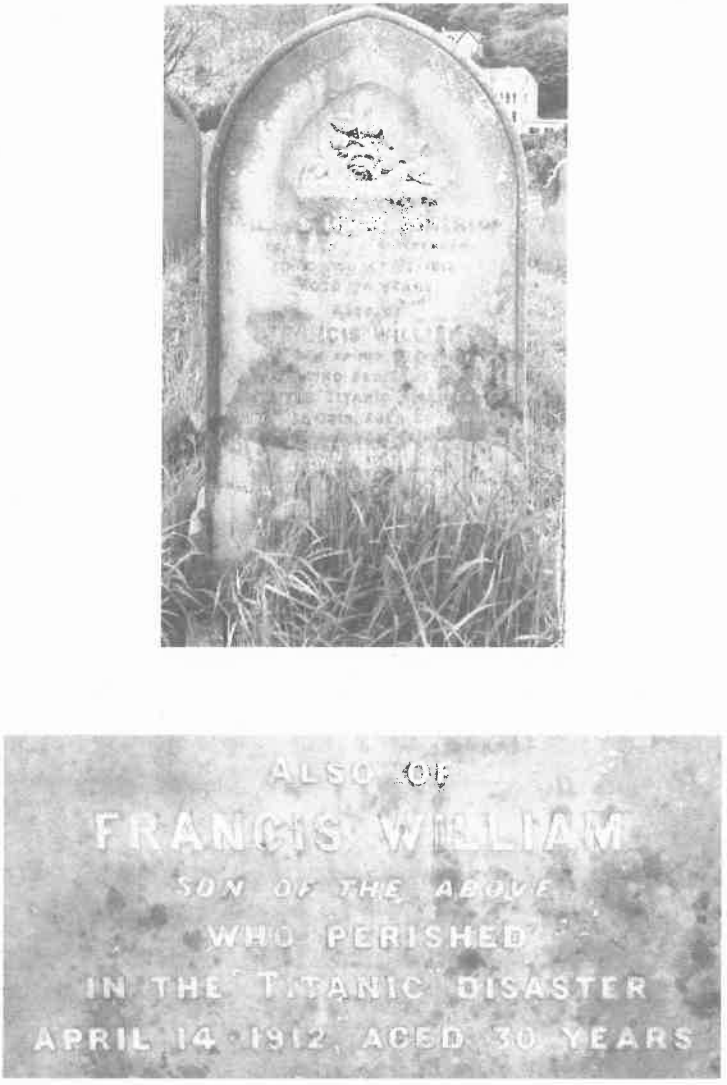


Figure 6.1
The memorial to a Francis William Somerton in the churchyard in Ilfracombe. It reads, 'Also of Francis William, son of the above, who perished in the Titanic disaster April 14 1912, aged 30 years'.

on some cabin numbers.[1] The response variable is an indicator for whether they survived (1) or not (0).

For the Analysis, it is crucial to split the data into a training set used to build the algorithm, and a test set that is kept apart and only used to assess performance – it would be serious cheating to look at the test set before we are ready with our algorithm. Like the Kaggle competition, we will take a random sample of 897 cases as our training set, and the remaining 412 individuals will comprise the test set.

This is a real, and hence fairly messy, data set, and some pre-processing is required. Eighteen passengers have missing fare information, and they have been assumed to have paid the median fare for their class of travelling. The number of siblings and parents have been added to create a single variable that summarizes family size. Titles needed simplifying: 'Mlle' and 'Ms' have been recoded as 'Miss', 'Mme' as 'Mrs', and a range of other titles are all coded as 'Rare titles'.*

It should be clear that, apart from the coding skills required, considerable judgement and background knowledge may be needed in simply getting the data ready for analysis, for example using any available cabin information to determine position on the ship. No doubt I could have done this better.

Figure 6.2 shows the proportion of different categories of passenger that survived, for the 897 passengers in the training set. All of these features have predictive ability on their own, with higher survival rates among passengers who are travelling in a better class of the ship, are female, children, paid more for their ticket, had a moderate size family, and had the title

* These include Dona, Lady, Countess, Capt, Col, Don, Dr, Major, Rev., Sir, Jonkheer.

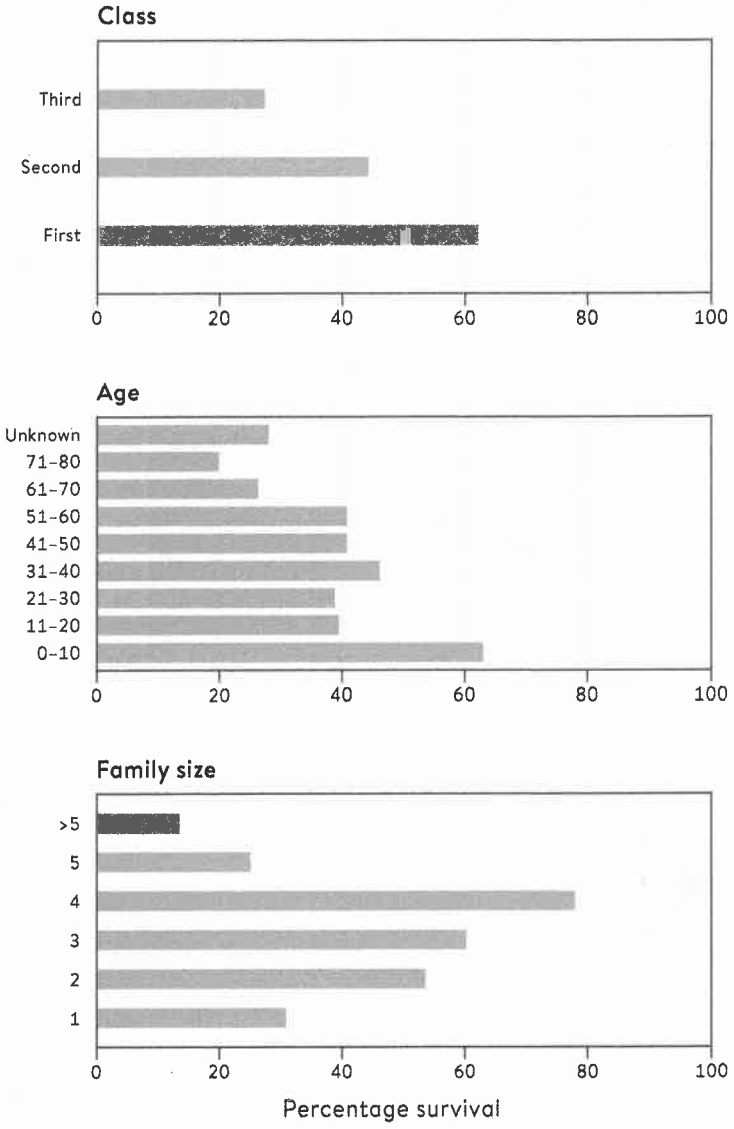
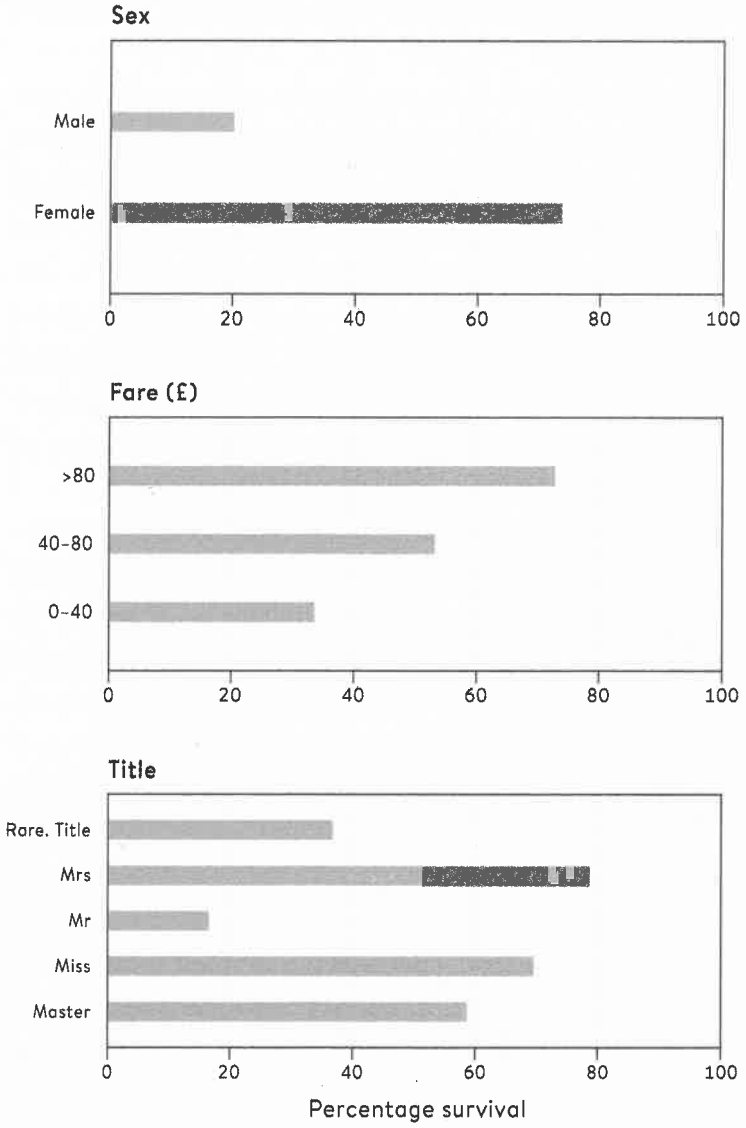


Figure 6.2
Summary survival statistics for training set of 897 *Titanic* passengers, showing the percentage of different categories that survived.


Sex
Male
Female
0
20
40
60
80
100
Fare (£)
>80
40-80
0-40
0
20
40
60
80
100
Title
Rare. Title
Mrs
Mr
Miss
Master
0
20
40
60
80
100
Percentage survival

Mrs, Miss, or Master. All of this matches what we might already suspect.

But these features are not independent. Better-class passengers presumably paid more for their tickets, and may be expected to be travelling with fewer children than would poorer emigrants. Many men were travelling on their own. And the specific coding may be important: should age be considered as a categorical variable, banded into the categories shown in Figure 6–2, or a continuous variable? Competitors have spent a lot of time looking at these features in detail and coding them up to extract the maximum information, but we shall instead proceed straight to making predictions.

Suppose we made the (demonstrably incorrect) prediction that 'Nobody survived'. Then, since 61% of the passengers died, we would get 61% right in the training set. If we used the slightly more complex prediction rule, 'All women survive and no men survive', we would correctly classify 78% of the training set. These naïve rules serve as good baselines from which to measure any improvements obtained from more sophisticated algorithms.

Classification Trees

A **classification tree** is perhaps the simplest form of algorithm, since it consists of a series of yes/no questions, the answer to each deciding the next question to be asked, until a conclusion is reached. Figure 6.3 displays a classification tree for the *Titanic* data, in which passengers are allocated to the majority outcome at the end of the branch. It is easy to see the factors that have been chosen, and the final conclusion. For example, Francis Somerton was titled 'Mr' in the

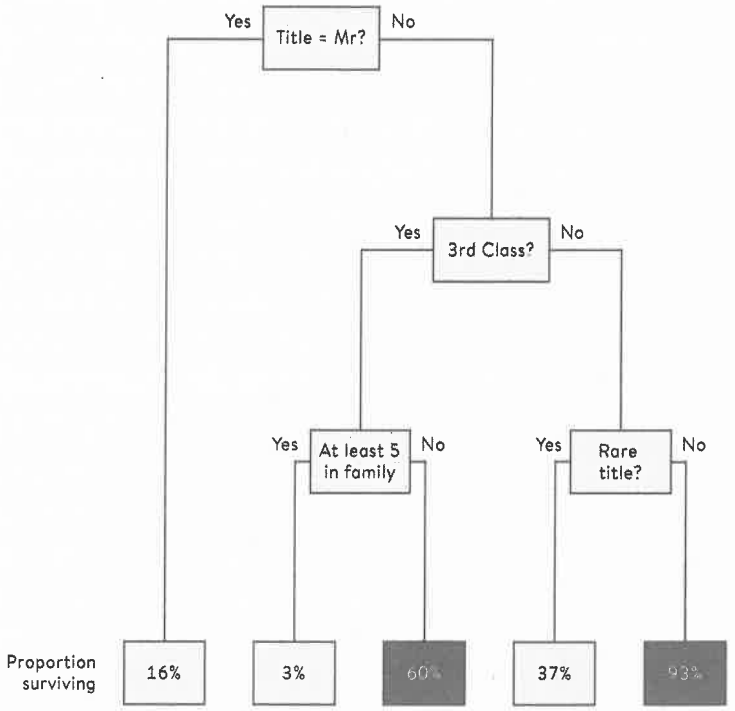


Figure 6.3
A classification tree for the *Titanic* data in which a sequence of questions leads a passenger to the end of a branch, at which point they are predicted to survive if the proportion of similar people in the training set who survived is greater than 50%; these surviving proportions are shown at the bottom of the tree. The only people predicted to survive are third-class women and children from smaller families, and all women and children in first and second class, provided they do not have rare titles.

database, and so would take the first left-hand branch. The end of this branch contains 58% of the training set, of which 16% survive. We could therefore assess, based on limited information, that Somerton had a 16% chance of surviving. Our simple algorithm identifies two groups with more than 50% survivors: women and children in first and second classes (as long they do not have a rare title), 93% of whom survive. And women and children from third class, provided they come from large families, in which case 60% survive.

Before seeing how such a tree is actually constructed, we need to decide what performance measures to use in our competition.

Assessing the Performance of an Algorithm

If algorithms are going to compete to be the most accurate, someone has to decide what 'accurate' means. In Kaggle's *Titanic* challenge this is simply the percentage of passengers in the test set that are correctly classified, and so after competitors build their algorithm, they upload their predictions for the response variable in the test set and Kaggle measures their accuracy.* We will present results for the whole test set at once (emphasizing that this is not the same as the Kaggle test set).

The classification tree shown in Figure 6.3 has an accuracy of 82% when applied to the training data on which it was

* In order not to have to wait until the end of the competition (in 2020 for the *Titanic* data) before anyone gets any feedback, Kaggle splits the test set into public and private sets. Competitors' accuracy scores on the public set are published on a leader board, and this provides a provisional ranking for all to see. But the performance on the private set is what is actually used to evaluate the final ranking of the competitors when the competition closes.

developed. When the algorithm is applied to the test set the accuracy drops slightly to 81%. The numbers of the different types of errors made by the algorithm are shown in Table 6.1 – this is termed the **error matrix**, or sometimes the confusion matrix. If we are trying to detect survivors, the percentage of true survivors that are correctly predicted is known as the **sensitivity** of the algorithm, while the percentage of true non-survivors that are correctly predicted is known as the **specificity**. These terms arise from medical diagnostic testing.

Although the overall accuracy is simple to express, it is a very crude measure of performance and takes no account of the confidence with which a prediction is made. If we look at the tips of the branches of the classification tree, we can see that the discrimination of the training data is not perfect, and at all branches there are some who survive and some not. The crude allocation rule simply chooses the outcome in the majority, but instead we could assign to new cases a *probability* of surviving corresponding to the proportion in the training set. For example, someone with the title 'Mr' could be given a probability of 16% of surviving, rather than a simple categorical prediction that they will not survive.

Algorithms that give a probability (or any number) rather than a simple classification are often compared using **Receiver Operating Characteristic (ROC) curves**, which were originally developed in the Second World War to analyse radar signals. The crucial insight is that we can vary the threshold at which people are predicted to survive. Table 6.1 shows the effect of using a threshold of 50% to predict someone a 'survivor', giving a specificity and sensitivity in the training set of 0.84 and 0.78 respectively. But we could have demanded a higher probability

	TRAINING SET			TEST SET		
	Predicted not to survive	Predicted to survive		Predicted not to survive	Predicted to survive	
Did not survive	475	93	568	228	45	273
Survived	71	258	329	35	104	139
	546	351	897	263	149	412
	Accuracy = (475 + 258)/897 = 82%			Accuracy = (228 + 104)/412 = 81%		
	Sensitivity = 258/329 = 78%			Sensitivity = 104/139 = 75%		
	Specificity = 475/568 = 84%			Specificity = 228/273 = 84%		

Table 6.1
Error matrix of classification tree on training and test data, showing accuracy (% correctly classified), sensitivity (% of survivors correctly classified) and specificity (% of non-survivors correctly classified).

in order to predict someone survives, say 70%, in which case the specificity and sensitivity would have been 0.98 and 0.50 respectively – with this more stringent threshold, we only identify half the true survivors but make very few false claims of surviving. By considering all possible thresholds for predicting a survivor, the possible values for the specificity and sensitivity form a curve. Note that the specificity axis conventionally decreases from 1 to 0 when drawing an ROC curve.

Figure 6.4 shows the ROC curves for training and test sets. A completely useless algorithm that assigns numbers at random would have a diagonal ROC curve, whereas the best algorithms will have ROC curves that move towards the top-left corner. A standard way of comparing ROC curves is by measuring the area underneath them, right down to the horizontal – this will be 0.5 for a useless algorithm, and 1 for a perfect one that gets everyone right. For our *Titanic* test set data, the area under the ROC curve is 0.82. It turns out that there is an elegant interpretation of this area: if we pick a true survivor and a true non-survivor at random, there is an 82% chance that the algorithm gives the true survivor a higher probability of surviving than the true non-survivor. Areas above 0.8 represent fairly good discriminatory ability.

The area under the ROC curve is one way of measuring how well an algorithm splits the survivors from the non-survivors, but it does not measure how good the probabilities are. And the people who are most familiar with probabilistic predictions are weather forecasters.

Suppose we want to predict whether or not it will rain tomorrow at a particular time and place. Basic algorithms

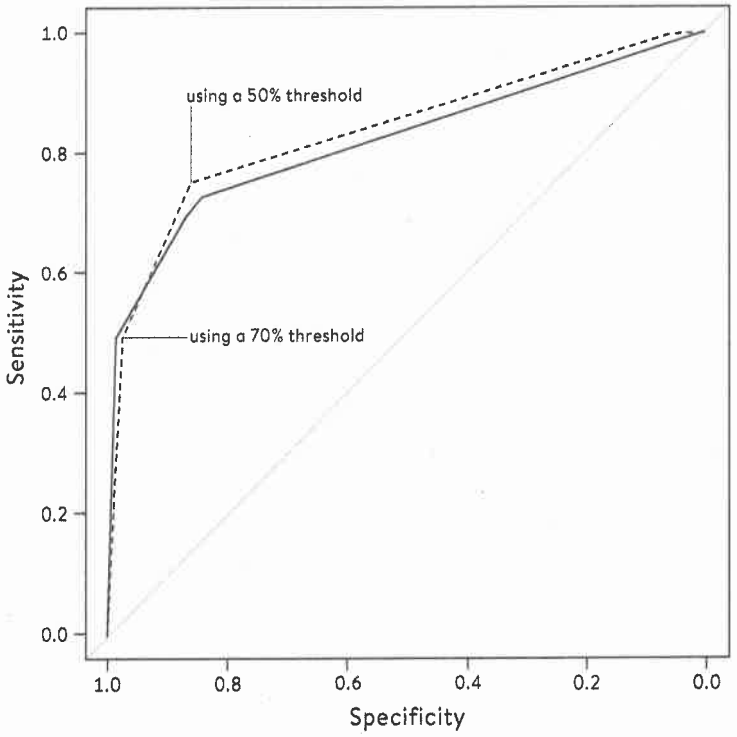


Figure 6.4
ROC curves for the classification tree of Figure 6.3 applied to training (dashed line) and test (solid line) sets. 'Sensitivity' is the proportion of survivors correctly identified. 'Specificity' is the proportion of non-survivors correctly labelled as not surviving. Areas under curves are 0.84 and 0.82 for training and test sets respectively.

How do we know how good 'probability of precipitation' forecasts are?

might simply produce a yes/no answer, which might end up being right or wrong. More sophisticated models might produce a probability of it raining, which allows more fine-tuned judgements – the action you take if the algorithm says a 50% chance of rain might be rather different than if it says 5%.

In practice weather forecasts are based on extremely complex computer models which encapsulate detailed mathematical formulae representing how weather develops from current conditions, and each run of the model produces a deterministic yes/no prediction of rain at a particular place and time. So to produce a **probabilistic forecast**, the model has to be run many times starting at slightly adjusted initial conditions, which produces a list of different 'possible futures', in some of which it rains and in some it doesn't. Forecasters run an 'ensemble' of, say, fifty models, and if it rains in five of those possible futures in a particular place and time, they claim a 'probability of precipitation' of 10%.

But how do we check how good these probabilities are? We cannot create a simple error matrix as in the classification tree, since the algorithm is never declaring categorically whether it will rain or not. We can create ROC curves, but these only examine whether days when it rains get higher predictions than when it doesn't. The critical insight is that we also need **calibration**, in the sense that if we take all the days in which the forecaster says 70% chance of rain, then it really

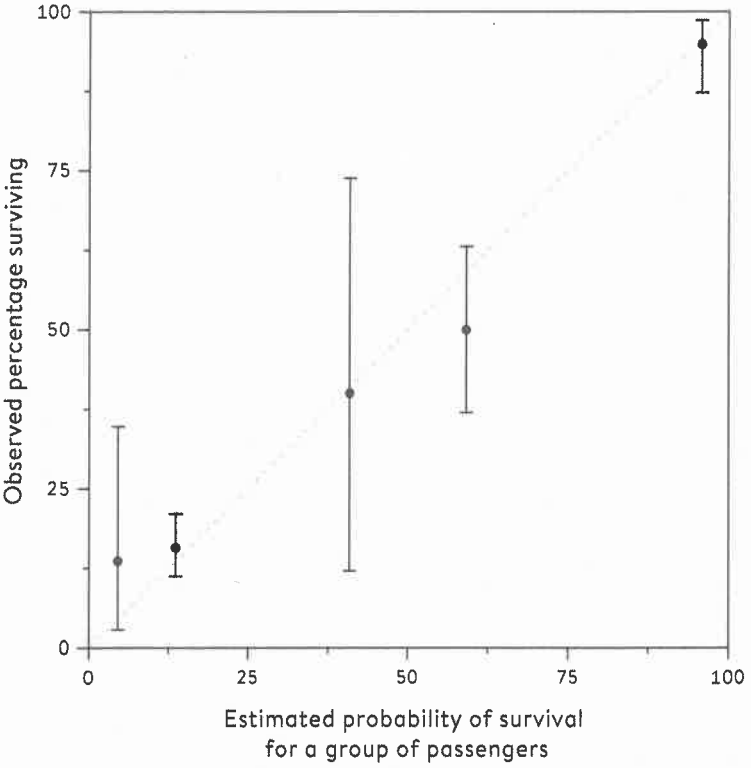


Figure 6.5
Calibration plot for the simple classification tree that provides probabilities of surviving the *Titanic* sinking, in which the observed proportion of survivors on the *y*-axis is plotted against the predicted proportion on the *x*-axis. We want the points to lie on the diagonal, showing the probabilities are reliable and mean what they say.

should rain on around 70% of those days. This is taken very seriously by weather forecasters – probabilities should mean what they say, and not be either over- or under-confident.

Calibration plots allow us to see how reliable the stated probabilities are, by collecting together, say, the events given a particular probability of occurrence, and calculating the proportion of such events that actually occurred.

Figure 6.5 shows the calibration plot for the simple classification tree applied to the test set. We want the points to lie near the diagonal since that is where the predicted probabilities match the observed percentages. The vertical bars indicate a region in which we would, given reliable predicted probabilities, expect the actual proportion to lie in 95% of cases. If these include the diagonal line, as in Figure 6.5, we can consider our algorithm to be well calibrated.

A Combined Measure of 'Accuracy' for Probabilities

While the ROC curve assesses how well the algorithm splits the groups, and the calibration plot checks whether the probabilities mean what they say, it would be best to find a simple composite measure that combines both aspects into a single number we could use to compare algorithms. Fortunately weather forecasters back in the 1950s worked out exactly how to do this.

If we were predicting a numerical quantity, such as the temperature at noon tomorrow in a particular place, the accuracy would usually be summarized by the error – the difference between the observed and predicted temperature. The usual summary of the error over a number of days is the **mean-squared-error** (**MSE**) – this is the average of the

squares of the errors, and is analogous to the least-squares criterion we saw used in regression analysis.

The trick for probabilities is to use the same mean-squared-error criterion as when predicting a quantity, but treating a future observation of 'rain' as taking on the value 1, and 'no rain' as being 0. Table 6.2 shows how this would work for a fictitious forecasting system. On Monday a probability of 0.1 is given to rain, but it turns out to be dry (true response is 0), and so the error is 0 – 0.1 = – 0.1. This is squared to give 0.01, and so on across the week. Then the average of these squared errors, B = 0.11, is a measure of the forecaster's (lack of) accuracy.* The average mean-squared-error is known as the **Brier score**, after meteorologist Glenn Brier, who described the method in 1950.

Unfortunately the Brier score is not easy to interpret on its own, and so it is difficult to get a feeling of whether any forecaster is doing well or badly; it is therefore best to compare it with a reference score derived from historical climate records. These 'climate-based' forecasts take no notice whatever of current conditions and simply state the probability of precipitation as the proportion of times in climate history in which it rained on this day. Anyone can make this forecast without any skill whatsoever – in Table 6.2 we assume this means quoting a 20% probability of rain for every day that week. This gives a Brier score for climate (which we call BC) of 0.28.

---

* It might be tempting to use the 'absolute error', meaning you would lose 0.1 when giving a 10% probability to an event that does not happen, as opposed to the squared error of 0.01. This apparently innocuous choice would be a big, big mistake. Some fairly basic theory shows that this 'absolute' penalty would lead people to rationally exaggerate their confidence in order to minimize their expected error, and state '0%' chance of rain, even if they genuinely thought the probability was 10%.

	Monday	Tuesday	Wednesday	Thursday	Friday	Mean-squared-error (Brier Score)
'Probability of precipitation'	0.1	0.2	0.5	0.6	0.3	
Did it actually rain?	No	No	Yes	Yes	No	
True response	0	0	1	1	0	
Error	–0.1	–0.2	0.5	0.4	–0.3	
Squared error	0.01	0.04	0.25	0.16	0.09	B = 0.54/5 = 0.11
Probability from climate	0.2	0.2	0.2	0.2	0.2	
Climate error	–0.2	–0.2	0.8	0.8	0.2	
Squared climate error	0.04	0.04	0.64	0.64	0.04	BC = 1.4/5 = 0.28

Table 6.2
Fictional 'probability of precipitation' forecasts of whether it will rain or not at midday next day at a specific location, with the observed outcome: 1 = did rain, 0 = did not rain. The 'error' is the difference between the predicted and observed outcome, and the mean-squared-error is the Brier score (B). The climate Brier score (BC) is based on using simple long-term average proportions of rain at this time of year as probabilistic forecasts, in this case assumed to be 20% for all days.

Any decent forecasting algorithm should perform better than predictions based on climate alone, and our forecast system has improved the score by BC – B = 0.28 – 0.11 = 0.17. Forecasters then create a 'skill score', which is the proportional reduction of the reference score: in our case, 0.61,* meaning our algorithm has made a 61% improvement on a naïve forecaster who uses only climate data.

Clearly our target is 100% skill, but we would only get this if our observed Brier score is reduced to 0, which only happens if we exactly predict whether it will rain or not. This is expecting rather a lot of any forecaster, and in fact skill scores for rain forecasting are now around 0.4 for the following day, and 0.2 for forecasting a week in the future.[2] Of course the laziest prediction is simply to say that whatever happened today will also happen tomorrow, which provides a perfect fit to historical data (today), but may not do particularly well in predicting the future.

When it comes to the *Titanic* challenge, consider the naïve algorithm of just giving everyone a 39% probability of surviving, which is the overall proportion of survivors in the training set. This does not use any individual data, and is essentially the equivalent to predicting weather using climate records rather than information on the current circumstances. The Brier score for this 'skill-less' rule is 0.232.

In contrast, the Brier score for the simple classification tree is 0.139, which is a 40% reduction from the naïve prediction, and so demonstrates considerable skill. Another way of

* The skill score is (BC–B)/BC = 1 – B/BC = 1 – 0.11/0.28 = 0.61

interpreting this Brier score of 0.139 is that it is exactly what would be obtained had you given all survivors a 63% chance of surviving, and all non-survivors a 63% chance of not surviving.

We shall see if we can improve on this score with some more complicated models, but first we need to issue a warning that they should not get *too* complicated.

Over-fitting

We do not need to stop at the simple classification tree shown in Figure 6.3. We could go on making the tree more and more complex by adding new branches, and this will allow us to correctly classify more of the training set as we identify more and more of its idiosyncrasies.

Figure 6.6 shows such a tree, grown to include many detailed factors. This has an accuracy on the training set of 83%, better than the smaller tree. But when we apply this algorithm to the test data its accuracy drops to 81%, the same as the small tree, and its Brier score is 0.150, clearly worse than the simple tree's 0.139. We have adapted the tree to the training data to such a degree that its predictive ability has started to decline.

This is known as **over-fitting**, and is one of the most vital topics in algorithm construction. By making an algorithm too complex, we essentially start fitting the noise rather than the signal. Randall Munroe (the cartoonist known for his *xkcd* comic strip) produced a fine illustration of over-fitting, by finding plausible 'rules' that US Presidents had followed, only for each to be broken at subsequent elections.[3] For example,

- 'No Republican has won without winning the House or Senate' – until Eisenhower did in 1952.

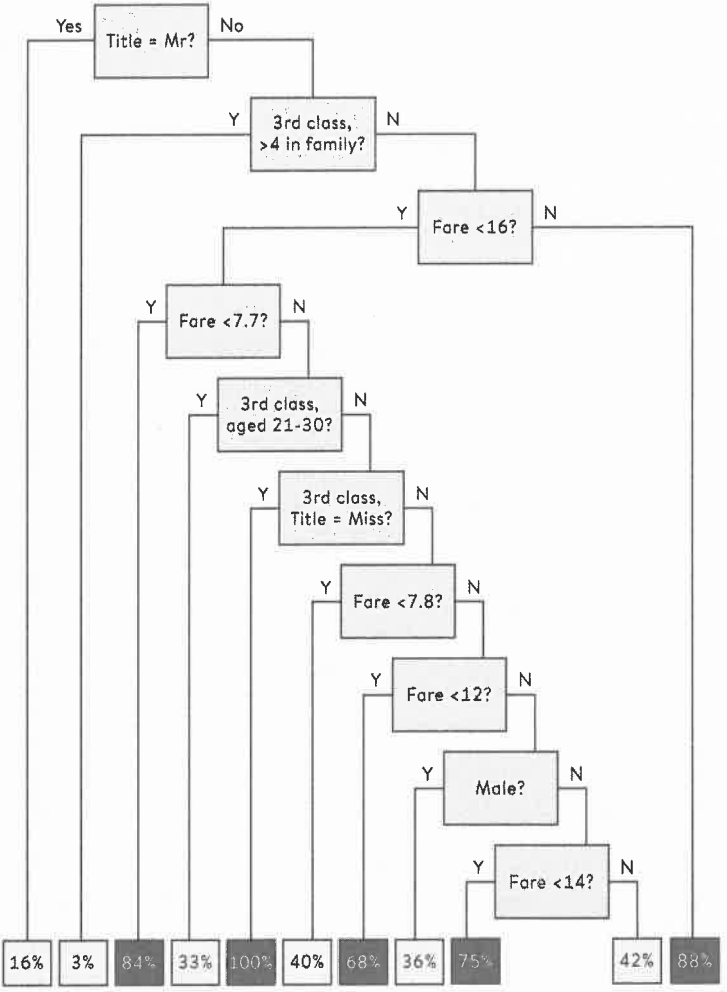


Figure 6.6
Over-fitted classification tree for the *Titanic* data. As in Figure 6.3, the percentage at the end of each branch is the proportion of passengers in the training set who survived, and a new passenger is predicted to survive if this percentage is greater than 50%. The rather strange set of questions suggests the tree has adapted too much to individual cases in the training set.

- 'Catholics can't win' – until Kennedy in 1960.
- 'No one has been elected President after a divorce' – until Reagan in 1980.

and so on, including some clearly over-refined rules such as

- 'No Democratic incumbent without combat experience has beaten someone whose first name is worth more in Scrabble' – until Bill (6 Scrabble points) Clinton beat Bob (7 Scrabble points) Dole in 1996.

We over-fit when we go too far in adapting to local circumstances, in a worthy but misguided effort to be 'unbiased' and take into account all the available information. Usually we would applaud the aim of being unbiased, but this refinement means we have less data to work on, and so the reliability goes down. Over-fitting therefore leads to less bias but at a cost of more uncertainty or variation in the estimates, which is why protection against over-fitting is sometimes known as the **bias/variance trade-off**.

We can illustrate this subtle idea by imagining a huge database of people's lives that is to be used to predict your future health – say your chance of reaching the age of eighty. We could, perhaps, look at people of your current age and socio-economic status, and see what happened to them – there might be 10,000 of these, and if 8,000 reached eighty, we might estimate an 80% chance of people like you reaching eighty, and be very confident in that number since it is based on a lot of people.

But this assessment only uses a couple of features to match you to cases in the database, and ignores more individual characteristics that might refine our prediction – for example

no attention is paid to your current health or your habits. A different strategy would be to find people who matched you much more closely, with the same weight, height, blood pressure, cholesterol, exercise, smoking, drinking, and so on and on: let's say we kept on matching on more and more of your personal characteristics until we narrowed it down to just two people in the database who were an almost perfect match. Suppose one had reached eighty and one had not. Would we then estimate a 50% chance of you reaching 80? That 50% figure is in a sense less biased, as it matches you so closely, but, because it is only based on two people, it is not a reliable estimate (i.e., it has large variance).

Intuitively we feel that there is a happy medium between these two extremes; finding that balance is tricky, but crucial. Techniques for avoiding over-fitting include regularization, in which complex models are encouraged but the effects of the variables are pulled in towards zero. But perhaps the most common protection is to use the simple but powerful idea of **cross-validation** when constructing the algorithm.

It is essential to test any predictions on an independent test set that was not used in the training of the algorithm, but that only happens at the end of the development process. So although it might show up our over-fitting at that time, it does not build us a better algorithm. We can, however, mimic having an independent test set by removing say 10% of the training data, developing the algorithm on the remaining 90%, and testing on the removed 10%. This is cross-validation, and can be carried out systematically by removing 10% in turn and repeating the procedure ten times, a procedure known as tenfold cross-validation.

All the algorithms in this chapter have some tunable parameters which are mainly intended to control the complexity of the final algorithm. For example, the standard procedure for building classification trees is to first construct a very deep tree with many branches that is deliberately overfitted, and then prune the tree back to something simpler and more robust: this pruning is controlled by a complexity parameter.

This complexity parameter can be chosen by the cross-validation process. For each of the ten cross-validation samples, a tree is developed for each of a range of different complexity parameters. For each value of the parameter, the average predictive performance over all the ten cross-validation test sets is calculated – this average performance will tend to improve up to a certain point, and then get worse as the trees become too complex. The optimal value for the complexity parameter is the one that gives the best cross-validatory performance, and this value is then used to construct a tree from the complete training set, which is the final version.

Tenfold cross-validation was used to select the complexity parameter in the tree in Figure 6.3, and to choose tuning parameters in all the models we consider below.

Regression Models

We saw in Chapter 5 that the idea of a regression model is to construct a simple formula to predict an outcome. The response variable in the *Titanic* data is a yes/no outcome indicating survival or not, and so a logistic regression is appropriate, just as for the child heart surgery data in Figure 5.2.

Table 6.3 shows the results from fitting a logistic regression. This has been trained using 'boosting', an iterative procedure designed to pay more attention to more difficult cases: individuals in the training set that are incorrectly classified at one iteration are given greater weight in the next iteration, with the number of iterations chosen using tenfold cross-validation.

The coefficients for the features of a particular passenger can be added up to give a total survival score. For example Francis Somerton would start with 3.20, subtract 2.30 for being in third class and 3.86 for being titled 'Mr', but then have 1.43 added back on for being a male in third class. He loses 0.38 for being in a family of one, giving a total score of – 1.91, which translates to a probability of 13% of surviving, slightly less than the 16% given by the simple classification tree.*

This is a 'linear' system, but note that **interactions** have been included which are essentially more complex, combined features, for example the positive score for the interaction of being in third class *and* a male helps counteract the extreme negative scores for the third class and 'Mr' already taken into account. Although we are focusing on predictive performance, these coefficients do provide some interpretation of the importance of different features.

Many more sophisticated regression approaches are available for dealing with large and complex problems, such as non-linear models and a process known as the LASSO, that simultaneously estimates coefficients and selects relevant

* To transform a total score S to a survival probability p, use the formula $p = 1/(1+e^{-S})$, where e is the exponential constant. This is the inverse of the logistic regression equation $\log_e p/(1-p) = S$.

Characteristic	Score
Starting score	3.20
Third class	–2.30
'Mr'	–3.86
Male in third class	+1.43
Rare Title	–2.73
Aged 51–60 in second class	–3.62
Each member of family	–0.38

Table 6.3
Coefficients applied to features in logistic regression for *Titanic* survivor data: negative coefficients decrease the chance of surviving, positive coefficients increase the chance.

predictor variables, essentially by estimating their coefficients to be zero.

More Complex Techniques

Classification trees and regression models arise from somewhat different modelling philosophies: trees attempt to construct simple rules that identify groups of cases with similar expected outcomes, while regression models focus on the weight to be given to specific features, regardless of what else is observed on a case.

The machine learning community makes use of classification trees and regressions, but has developed a wide range of alternative, more complex methods for developing algorithms. For example:

- *Random forests* comprise a large number of trees, each producing a classification, with the final classification decided by a majority vote, a process known as bagging.
- *Support vector machines* try to find linear combinations of features that best split the different outcomes.
- *Neural networks* comprise layers of nodes, each node depending on the previous layer by weights, rather like a series of logistic regressions piled on top of each other. Weights are learned by an optimization procedure, and, rather like random forests, multiple neural networks can be constructed and averaged. Neural networks with many layers have become known as deep-learning models: Google's Inception image-recognition system is said to have over twenty layers and over 300,000 parameters to estimate.

- *K-nearest-neighbour* classifies according to the majority outcome among close cases in the training set.

The results of applying some of these methods to the *Titanic* data, with tuning parameters chosen using tenfold cross-validation and ROC as an optimization criterion, are shown in Table 6.4.

The high accuracy of the naïve rule, 'All females survive, all males do not', which either beats or is close behind more complex algorithms, demonstrates the inadequacy of crude 'accuracy' as a measure of performance. The random forest produces the best discrimination reflected in the area under the ROC curve, although perhaps surprisingly the probabilities coming from the simple classification tree have the best Brier score. There is therefore no clear winning algorithm. Later, in Chapter 10, we shall check whether we can confidently claim there is a proper winner on any of these criteria, since the winning margins might be so small that it can be explained by chance variation – say in who happened to end up in the test and training set.

This reflects a general concern that algorithms that win Kaggle competitions tend to be very complex in order to achieve that tiny final margin needed to win. A major problem is that these algorithms tend to be inscrutable black boxes – they come up with a prediction, but it is almost impossible to work out what is going on inside. This has three negative aspects. First, extreme complexity makes implementation and upgrading a great effort: when Netflix offered a $1m prize for prediction recommendation systems, the winner was so complicated that Netflix ended up not using it. The second

Method	Accuracy (high is good)	Area under ROC curve (high is good)	Brier score (low is good)
Everyone has a 39% chance of surviving	0.639	0.500	0.232
All females survive, all males do not	0.786	0.578	0.214
Simple classification tree	**0.806**	0.819	**0.139**
Classification tree (over-fitted)	**0.806**	0.810	0.150
Logistic regression	0.789	0.824	0.146
Random forest	0.799	**0.850**	0.148
Support Vector Machine (SVM)	0.782	0.825	0.153
Neural network	0.794	0.828	0.146
Averaged neural network	0.794	0.837	0.142
K-nearest-neighbour	0.774	0.812	0.180

Table 6.4
The performance of different algorithms on *Titanic* test data: bold indicates the best results. Complex algorithms have been optimized to maximize the area under the ROC curve.

negative feature is that we do not know how the conclusion was arrived at, or what confidence we should have in it: we just have to take it or leave it. Simpler algorithms can better explain themselves. Finally, if we do not know how an algorithm is producing its answer, we cannot investigate it for implicit but systematic biases against some members of the community – a point I expand on below.

All this points to the possibility that quantitative performance may not be the sole criterion for an algorithm, and once performance is 'good enough', it may be reasonable to trade off further small increases for the need to retain simplicity.

Who was the luckiest person on the *Titanic*?

The survivor with the highest Brier score when averaged over all the algorithms might be considered the most surprising survivor. This was Karl Dahl, a 45-year-old Norwegian/Australian joiner travelling on his own in third class, who had paid the same fare as Francis Somerton; two algorithms even gave him a 0% chance of surviving. He apparently dived into the freezing water and clambered into Lifeboat 15, in spite of some on the lifeboat trying to push him back. Maybe he just used his strength.

This is in stark contrast to Francis Somerton from Ilfracombe, whose death, we have found, fitted into the general pattern. Rather than having a successful husband in America, his wife Hannah Somerton was left just £5, less than Francis spent on his ticket.

Challenges of Algorithms

Algorithms can display remarkable performance, but as their role in society increases so their potential problems become highlighted. Four main concerns can be identified.

- *Lack of robustness*: Algorithms are derived from associations, and since they do not understand underlying processes, they can be overly sensitive to changes. Even if we are only concerned with accuracy rather than scientific truth, we still need to remember the basic principles of the PPDAC cycle, and the stages of going from the data obtained from a sample through to statements being made about a target population. For predictive analytics, this target population comprises future cases, and if everything stays the same, then algorithms constructed on past data should perform well. But the world does not always stay the same. We've noted the failure of algorithms in the changing financial world of 2007–8, and another notable example was the attempt by Google to predict flu trends based on the pattern of search terms being submitted by users. This initially performed well but then in 2013 started to dramatically over-predict flu rates: one explanation is that changes introduced by Google into the search engine may have led to more search terms that pointed to flu.
- *Not accounting for statistical variability*: Automated rankings based on limited data will be unreliable. Teachers in the US have been ranked and penalized for the performance of their students in a single year,

although class sizes of less than thirty do not provide a reliable basis for assessing the value added by a teacher. This will reveal itself in teachers having implausibly dramatic changes in annual assessment: in Virginia, a quarter of teachers showed more than 40-point differences in a 1–100 scale from year-to-year.*

- *Implicit bias*: To repeat, algorithms are based on associations, which may mean they end up using features that we would normally think are irrelevant to the task in hand. When a vision algorithm was trained to discriminate pictures of huskies from German Shepherds, it was very effective until it failed on huskies that were kept as pets – it turned out that its apparent skill was based on identifying snow in the background.[4] Less trivial examples include an algorithm for identifying beauty that did not like dark skin, and another that identified Black people as gorillas. Algorithms that can have a major impact on people's lives, such as those deciding credit ratings or insurance, may be banned from using race as a predictor but might use postcodes to reveal neighbourhood, which is a strong proxy for race.
- *Lack of transparency*: Some algorithms may be opaque due to their sheer complexity. But even simple regression-based algorithms become totally inscrutable if their structure is private, perhaps through being a proprietary commercial product. This is one of the major complaints about so-called recidivism algorithms, such

* From Cathy O'Neil's book *Weapons of Math Destruction*, which provides many examples of the misuse of algorithms.

as Northpointe's Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) or MMR's Level of Service Inventory – Revised (LSI-R).[5] These algorithms produce a risk-score or category that can be used to guide probation decisions and sentencing, and yet the way in which the factors are weighted is unknown. Furthermore, since information about upbringing and past criminal associates is collected, decisions are not solely based on a personal criminal history but background factors that have been shown to be associated with future criminality, even if the underlying common factor is poverty and deprivation. Of course, if all that mattered was accurate prediction, then anything goes and any factor, even including race, might be used. But many argue that fairness and justice demand that these algorithms should be controlled, transparent and able to be appealed against.

Even for proprietary algorithms, some degree of explanation is possible provided we can experiment with different inputs. When purchasing online insurance, the quoted premium is calculated according to an unknown formula subject only to certain legal constraints: for example, in the UK car insurance quotes cannot take into account the gender of the applicant, life insurance cannot use race or any genetic information except Huntingdon's disease, and so on. But we can still get an idea of the influence of different factors by systematically lying and seeing how the quotation changes: this allows a certain degree of reverse-engineering of the algorithm to see what is driving the premium.

There is increasing demand for accountability of algorithms that affect people's lives, and requirements for comprehensible explanation of conclusions are being built into legislation. These demands militate against complex black boxes, and may lead to a preference for (rather old-fashioned) regression-based algorithms that make the influence of each item of evidence clear.

But having looked at the dark side of algorithms, it is fitting to end with an example that seems entirely beneficial and empowering.

> What is the expected benefit of adjuvant therapy following breast cancer surgery?

Nearly all women newly diagnosed with breast cancer will receive some form of surgery, although this might be limited in extent. A critical issue is then the choice of adjuvant therapy that follows surgery in order to reduce the chances of recurrence and subsequent death from breast cancer, and treatment options may include radiotherapy, hormone therapy, chemotherapy and other drug options. Within the PPDAC cycle, this is the Problem.

The Plan adopted by UK researchers was to develop an algorithm to help with this decision, using Data on 5,700 historical cases of women with breast cancer obtained from the UK Cancer Registry. The Analysis comprised the construction of an algorithm that would use detailed information on the woman and her tumour in order to calculate her chances of survival for up to ten years following surgery, and how these

changed with different treatments. But care is required in analysing the outcomes of women given these treatments in the past: they were given the treatments for unknown reasons and we cannot use the apparent benefits observed in the database. Instead a regression model is fitted, with survival as the outcome, but forcing the effect of treatments to be those estimated from reviews of large-scale clinical trials. The subsequent algorithm is publicly available, and its discrimination and calibration has been checked on independent data sets comprising 27,000 women.[6]

The resulting computer software is called Predict 2.1, and the results are Communicated through the proportions of similar women expected to survive five and ten years for different adjuvant treatments. Some results for a fictitious woman are shown in Table 6.5.

Predict 2.1 is not perfect, and the figures in Table 6.5 can only be used as ballpark guides for an individual: they are what we would expect to happen to women who match the features included in the algorithm, and additional factors should be taken into account for a specific woman. Nevertheless, Predict 2.1 is used routinely for tens of thousands of cases a month, both in multidisciplinary team meetings (MDTs) in which a patient's treatment options are formulated, and in communicating that information to the woman. For those women who wish to fully engage in their treatment choices, a process known as 'shared-care', it can provide information normally only available to the clinicians, and empower them to have greater control over their lives. The algorithm is not proprietary, the software is open source and the system is regularly being upgraded

Treatment	Additional benefit over previous treatments	Overall survival %
Surgery only	–	64%
+ Hormone therapy	7%	70%
+ Chemotherapy	6%	76%
+ Trastuzamab (Herceptin)	3%	79%
For women free from cancer		87%

Table 6.5
Using the Predict 2.1 algorithm, the proportion of 65-year-old women expected to survive ten years after surgery for breast cancer, when a 2cm grade 2 tumour was detected at screening, with two positive nodes, and ER, HER2 and Ki-67 status are all positive. The cumulative expected benefits for different adjuvant treatments are shown, although these treatments may have adverse effects. The surviving proportion for 'women free from cancer' represents the best survival achievable, given the age of the woman.

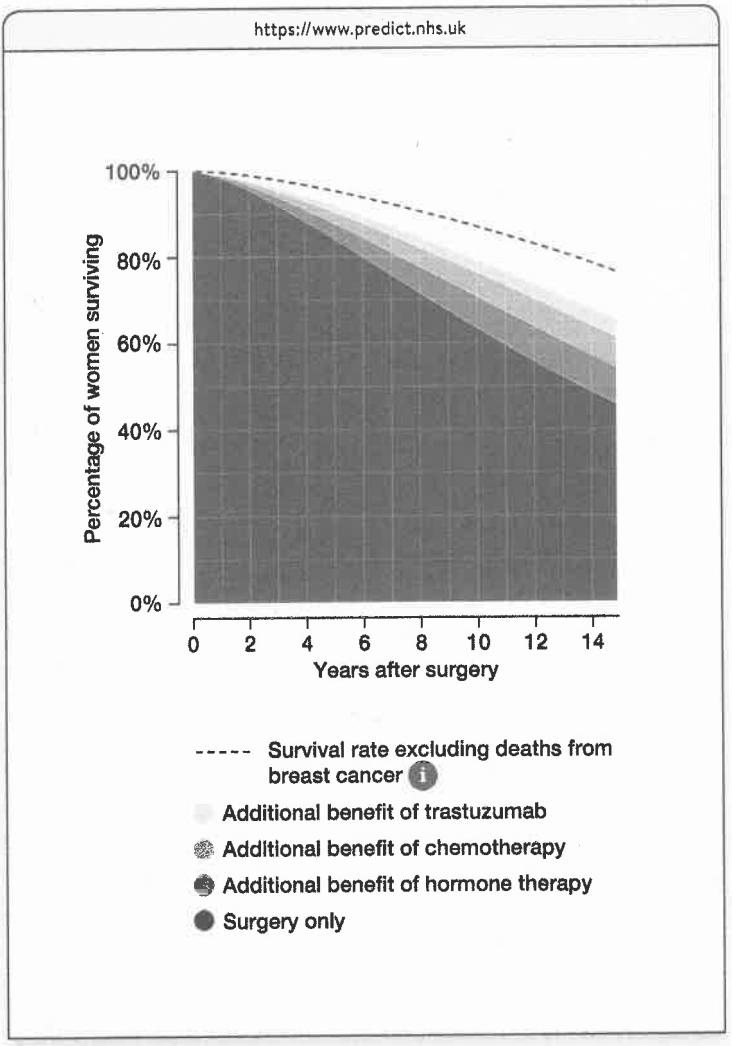


Figure 6.7
Survival curves from Predict 2.1 for up to fifteen years post-surgery, for women with the features listed in the legend to Table 6.5, showing the cumulative additional survival from further treatments. The area above the dashed line represents women with breast cancer who die of other causes.

to provide further information, including adverse effects of treatments.

Artificial Intelligence

Ever since its first use in the 1950s, the idea of artificial intelligence (AI) has received periodic hype and enthusiasm and subsequent troughs of criticism. I was working on computer-aided diagnosis and handling uncertainty in AI in the 1980s, when much of the discourse was framed in terms of a competition between approaches based on probability and statistics, those based on encapsulating expert 'rules' of judgement or those trying to emulate cognitive capacities through neural networks. The field has now matured, with a more pragmatic and ecumenical approach to its underlying philosophy, although the hype has not gone away.

AI comprises intelligence demonstrated by machines, which is a suitably wide-ranging idea. It is a much bigger topic than the restricted issue of algorithms discussed in this chapter, and statistical analysis is only one component to building AI systems. But, as demonstrated by the extraordinary recent achievements of algorithms in vision, speech, games and so on, statistical learning plays a major part in the successes in 'narrow' AI. Systems such as Predict, which previously would be thought of as statistics-based decision-support systems, might now reasonably be called AI.*

Many of the challenges listed above come down to algorithms only modelling associations, and not having an idea of

* Perhaps for no other reason than to attract funding . . .

underlying causal processes. Judea Pearl, who has been largely responsible for the increased focus on causal reasoning in AI, argues that these models only allow us to answer questions of the type, 'We have observed X, what do we expect to observe next?' Whereas general AI needs a causal model for how the world actually works, which would allow it to answer human-level questions concerning the effect of interventions ('What if we do X?'), and counterfactuals ('What if we hadn't done X?').

We are a long way from AI having this ability.

This book emphasizes the classic statistical problems of small samples, systematic bias (in the statistical sense) and lack of generalizability to new situations. The list of challenges for algorithms shows that although having masses of data may reduce the concern about sample size, the other problems tend to get worse, and we are faced with the additional problem of explaining the reasoning of an algorithm.

Having bucketloads of data only increases the challenges in producing robust and responsible conclusions. A basic humility when building algorithms is crucial.

Summary

- Algorithms built from data can be used for classification and prediction in technological applications.
- It is important to guard against over-fitting an algorithm to training data, essentially fitting to noise rather than signal.
- Algorithms can be evaluated by the classification accuracy, their ability to discriminate between groups, and their overall predictive accuracy.
- Complex algorithms may lack transparency, and it may be worth trading off some accuracy for comprehension.
- The use of algorithms and artificial intelligence presents many challenges, and insights into both the power and limitations of machine-learning methods is vital.

CHAPTER 6: ALGORITHMS, ANALYTICS AND PREDICTION

1. The *Titanic* data can be downloaded from http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic3.xls.
2. Verifying probability of precipitation: http://www.cawcr.gov.au/projects/verification/POP3/POP3.html.
3. 'Electoral Precedent', *xkcd*, https://xkcd.com/1122/.
4. http://innovation.uci.edu/2017/08/husky-or-wolf-using-a-black-box-learning-model-to-avoid-adoption-errors/.
5. The use of COMPAS and MMR algorithms is critiqued in C. O'Neil, *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy* (Penguin, 2016).
6. NHS, Predict: Breast Cancer (2.1): http://www.predict.nhs.uk/predict_v2.1/.

CHAPTER 7: HOW SURE CAN WE BE ABOUT WHAT IS GOING ON? ESTIMATES AND INTERVALS

1. UK labour market statistics, January 2018: https://www.ons.gov.uk/releases/uklabourmarketstatisticsjan2018. Bureau of Labor Statistics, 'Employment Situation Technical Note 2018', https://www.bls.gov/news.release/empsit.tn.htm.

CHAPTER 8: PROBABILITY – THE LANGUAGE OF UNCERTAINTY AND VARIABILITY

1. Consider Game 1. There are many ways of winning, but only one way of losing – throwing four non-sixes in a row. It is therefore easier to find the probability of losing (this is a common trick). The chance of throwing a non-six is $1 - \frac{1}{6} = \frac{5}{6}$ (complement rule), and the chance of throwing four non-sixes in a row is $\frac{5}{6} \times \frac{5}{6} \times \frac{5}{6} \times \frac{5}{6} = \left(\frac{5}{6}\right)^4 = \frac{625}{1296} = 0.48$ (multiplication rule). So the probability of winning is $1 - 0.48 = 0.52$ (complement rule again). Similar reasoning for Game 2 leads to the probability of winning to be $1 - \left(\frac{35}{36}\right)^{24} = 0.49$, showing that Game 1 was slightly more favourable. These rules also show the error in the Chevalier's reasoning – he was adding the probabilities of events that were not mutually exclusive. By his reasoning