THE RISE OF Artificial Intelligence

Real-world Applications for Revenue and Margin Growth

Zbigniew Michalewicz
Leonardo Arantes
Matt Michalewicz

Published by Hybrid Publishers
Melbourne Victoria Australia

Hybrid Publishers,
PO Box 52, Ormond VIC Australia 3204.
www.hybridpublishers.com.au

NATIONAL LIBRARY OF AUSTRALIA
A catalogue record for this book is available from the National Library of Australia

First published 2021
Text design & layout by Midland Typesetters, Australia
Cover design by Marchese Design
ISBN 9781925736625 (p)
Printed by Tingleman Pty Ltd

*Dedicated to the entrepreneurs, scientists, and business leaders
that have paved the way for Artificial Intelligence over the decades past,
and are paving the way for its future in the decades to come.*

PREFACE

What This Book is About and How to Read It

"We're at the beginning of a golden age of AI. Recent advancements have already led to inventions that previously lived in the realm of science fiction—and we've only scratched the surface of what's possible."

Jeff Bezos, *Amazon CEO*

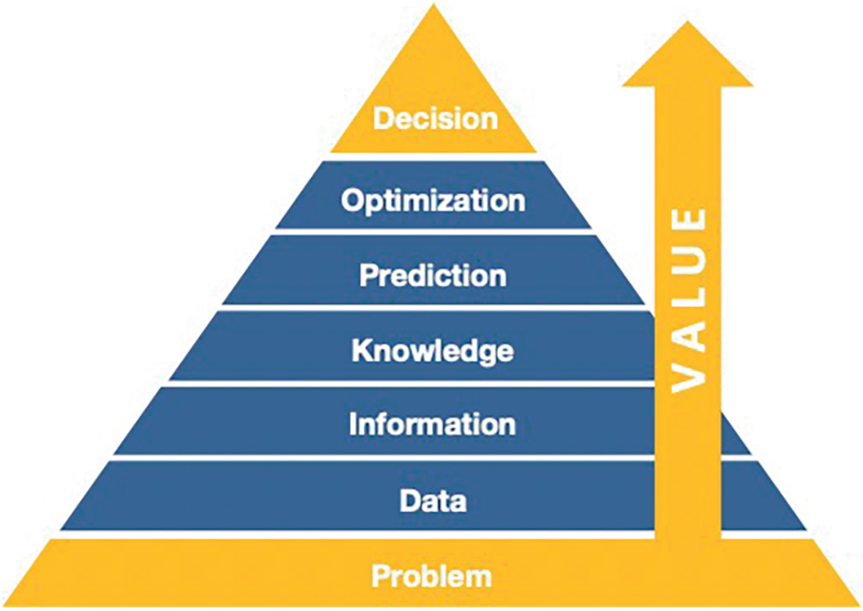
Few terms have captured our imagination in recent times like "Artificial Intelligence." And not just through sensationalized media articles about how AI will soon displace all jobs and rule the world, but also through movies, books, and television shows. It now seems that everyone "knows" about AI; that everyone has an opinion. And yet, in our experience, few people actually understand what Artificial Intelligence is and isn't, where the field came from and where it's heading, and how the technology can be harnessed to generate commercial outcomes.

Given the immense amount of disinformation and misunderstanding, we have written this book to demystify the subject of AI and explain it in simple language. Most importantly, we have written this book with the business manager in mind, someone interested in the topic from a real-world, commercial perspective—a perspective of how the technology can create value and increase competitiveness *today*, rather than what might happen in 25 years' time or how a superior intelligence might overcome the human race in the distant future. Such philosophical treatises are thought-provoking (to say the least) and the subject of many books published each year, but this isn't one of them. Instead, *The Rise of Artificial Intelligence* provides a commercial exploration of AI, with particular emphasis on how AI-based systems can improve decision making in organizations of all shapes and sizes.

As such, this book presents Artificial Intelligence through the lens of decision making for two reasons: First, because the world has reached a level of such unprecedented speed, complexity, and noise, that no one can assess and evaluate all the available data when making decisions; and secondly, because the decisions we make affect the outcomes we achieve. In other words, better business decisions lead to better business outcomes. Although Artificial

Intelligence can be applied to many areas besides decision making—such as automation and robotics, or image and speech recognition—these subjects don't feature heavily in the pages ahead except for Chapter 1, where we provide an overview of the research areas of AI. Ultimately, revenue and margin growth comes down to the decisions an organization makes (or doesn't make), and hence the application of AI to decision making is our primary focus.

To best present the concepts in this book, we've used a *problem-to-decision pyramid* to represent the continuum that exists in terms of an organization's ability to improve its decision making:



Each layer of this pyramid represents a step in the journey for improved decision making: the higher we go, the better our decisions (and the more value we can create). The structure of *The Rise of Artificial Intelligence* reflects the structure of this pyramid, with the first two parts of the book investigating each layer of the pyramid, and the last two parts illustrating the application of Artificial Intelligence to real-world problems for the purpose of generating revenue and margin growth.

Chapter 1 begins with a high-level overview of Artificial Intelligence—its history, areas of research, and current progress and challenges—before introducing the *problem-to-decision pyramid* in Chapter 2, which conceptualizes the journey from defining a problem to making a decision through the use of data, information, knowledge, prediction, and optimization. Chapter 3 concludes Part 1with an in-depth examination of a complex business problem set in the fast-moving consumer goods industry, which is used to explain the role of objectives, business rules and constraints, and the application of Artificial Intelligence algorithms for improved decision making.

This complex business problem of promotional planning and pricing is then used as a running example throughout Part II, which explores the inner workings of predictive models, optimization methods, and various learning algorithms. Because data and modeling form the basis of prediction and optimization, this part of the book opens with a chapter on data and modeling, along with a discussion of common issues such as data availability, completeness, and preparation. In Chapters 5 and 6 we review various AI and non-AI methods for predictive modeling and optimization, whereas in Chapter 7 we present adaptability and learning concepts—which together (i.e. prediction, optimization, and self-learning) comprise the backbone of any AI-based software system.

As an important aside, Chapters 4 through 7 represent the most technical material of the entire book, attempting to explain the innermost mechanics of several Artificial Intelligence algorithms such as neural networks and genetic programming. Although non-technical readers can easily progress through Part II to gain a deeper understanding of algorithms and models, readers without an interest in data, problem modeling, or how Artificial Intelligence algorithms work, can jump straight to Part III, which presents real-world applications of Artificial Intelligence.

The application areas in Part III explore the problem-to-decision pyramid in the context of real-world problems and business objectives, covering both the lower layers of the pyramid focusing on data and the analytical landscape of an organization (i.e. information and knowledge), as well as the upper layers of prediction, optimization, and self-learning, and how they're enabled by Artificial Intelligence methods. For ease of reading, we've divided Part III into three chapters, each being dedicated to a specific business function—in particular, *sales, marketing,* and *supply chain*. These case studies are based on an enterprise software platform called Decision Cloud®, which is a modularized, cloud-based platform that empowers staff to make better and faster decisions through the use of Artificial Intelligence.

And finally, Part IV concludes the book with common questions and concerns that organizations have on the application of Artificial Intelligence, such as: *"Would AI work for me?"* and *"Where should I start?"* These two chapters provide practical advice for selecting the right business problem, developing a business case, choosing a technology partner, as well as other topics such as digitalization and change management.

To improve the reader's understanding of the content, we've also created a set of supplementary videos that can be accessed at: www.Complexica.com/book/RiseofAI/. These videos bring to life the concepts presented in each chapter—for example, by providing a visual explanation of ant system algorithms in Chapter 1, the layers of the problem-to-decision pyramid in

Chapter 2, the workflow of promotional planning and pricing in Chapter 3, and so on. In these videos we're able to "show" concepts that can only be "told" within the confines of the printed page.

In terms of how to read this book or watch the videos, the ideal way is to progress sequentially from Chapter 1 to 12. For the less technically-inclined reader, however it's possible to jump around in any sequence that best satisfies curiosity and interest. For example, the reader might begin with an overview of Artificial Intelligence in Chapter 1, then progress to the application areas in Chapters 8, 9, and 10, before returning to Chapters 2 and 3 to better appreciate the problem-to-decision pyramid and the intricacies of solving complex business problems (after all, why are complex business problem so difficult to solve?). Alternatively, a reader might start with the application areas in Chapters 8, 9, and 10, then move back into Part II to better understand how algorithms and models work, before progressing to Part IV for practical advice for initiating an Artificial Intelligence project.

However, regardless of the reader's technical sophistication or their interest in the implementation aspects of AI-based software, it's highly recommended that everyone start with the first two chapters for an introduction into the world of Artificial Intelligence and an overview of basic concepts and terminology. From this perspective, the sequence of reading the remaining chapters is of far lesser importance.

Lastly, we'd like to say that the material presented in this book is the result of 40 years of first-hand Artificial Intelligence research within university settings, and more than twenty years of implementing AI-based enterprise software systems in many (often very large[1]) organizations across three continents. With that in mind, we'd like to thank everyone who made this book possible, with our special appreciation going to many Australian companies we collaborated with over the years in the application of Artificial Intelligence, such as PFD Foods, BHP Billiton, BMA, Pernod Ricard Winemakers, Lion Drinks, Bunzl, DuluxGroup, Rio Tinto, Metcash, Pfizer, Janssen, Haircare Australia, Fortescue Metals Group, CBH Group, Roy Hill, Glencore, Polyaire, Treasury Wine Estates, and Costa Group. Within these companies, we'd like to thank Chris Baddock, John Barakat, Renato Bellon, Simon Bennett, Damian Bourne, Warren Brodie, Michael Brooks, Pierre-Yves Calloc'h, Daryl Chim,

[1] Our experiences of implementing enterprise-grade software based on the latest Artificial Intelligence algorithms and methods are based on many projects with global giants—such as BHP Billiton, General Motors, Bank of America, Pernod Ricard, Unilever, Air Liquide, Ford Motor Company, Glencore, Beiersdorf, Rio Tinto, and ChevronTexaco, among many others—as well as smaller companies that benefited from the research & development and innovation carried out by these larger organizations.

Richard Cohen, Jevan Dickinson, Andrew Endicott, Eglantine Etiemble, Scott Fellingham, Greg Feutrill, Garth Gauvin, Ward Gauvin, Scott Graham, Chris Green, Kylie Grigg, Richard Hansen, Mark Hayden, Kim Heatherton, Mark Ivory, James Jones, Mike Lomman, Brett McKinnon, Stuart McNab, Doug Misener, Luke Mitchell, Stephen Mooney, Aemel Nordin, Mark Powell, Rod Pritchard, Robin Pyne, Mathew Regan, Darryl Schafferius, Mark Shephard, Jon Simpson, Kerry Smith, Richard Taylor, Soner Teknikeller, Lance Ward, John Warda, and Joel Zamek.

We'd also like thank a few individuals who contributed to the content and ideas in this book, namely, Reza Bonyadi, Łukasz Brocki, Tom Heyworth, Xiang Li, Łukasz Olech, Ali Shemshadi, Larisa Stamova, Chris Zhu, as well as members of Complexica's scientific advisory board who we've worked with over the years: Reza Bonyadi, Łukasz Brocki, Longbing Cao, Raymond Chiong, Vic Ciesielski, Carlos Coello, Ernesto Costa, Kalyanmoy Deb, Kenneth De Jong, A.E. Eiben, Xiaodong Li, Masoud Mohammadian, Pablo Moscato, Frank Neumann, Zbigniew Raś, Markus Wagner, Thomas Weise, Adam Wierzbicki, and Mengjie Zhang.

And finally, it was a great pleasure to write about a topic that's been the central focus of our working lives for so many years, and we hope that readers enjoy this book as much as we enjoyed writing it. We believe that anyone in any organization who makes operational, tactical, or strategic decisions—whether on the factory floor or in the boardroom—will find this book valuable for understanding the science and technology behind better decisions. Enjoy!

Adelaide, Australia
March 2021

Zbigniew Michalewicz
Leonardo Arantes
Matt Michalewicz

TABLE OF CONTENTS

CHAPTER 2

Complex Business Problems

"The whole universe sat there, open to the man
who could make the right decisions."
Frank Herbert, *Dune*

Recent years have seen terms like *data science*, *algorithms*, *machine learning*, and *big data* solidify their position in our everyday vocabulary, with articles on Artificial Intelligence becoming commonplace in business and mainstream publications. With the growing popularity of websites that make recommendations and smartphones that take voice commands, there is a growing appreciation for how AI-enabled functionality adds value to our day-to-day lives. Parallel to that, in the enterprise space, there is likewise a growing trend of embedding AI functionality into Customer Relationship Management (CRM), Enterprise Resource Planning (ERP), and other corporate systems, so they can handle more sophisticated workflows and deliver more value. In this context, it should come as no surprise that organizations of all shapes and sizes are increasingly asking: *What is Artificial Intelligence truly capable of*? *What is it best suited for*? *What could that mean for my organization*?

A good place to start in answering such questions lies in our day-to-day usage of Artificial Intelligence, most likely through mobile apps that recommend books, movies, the best route through traffic, even food we might come to love, despite having never tried it before. What these applications have in common is the use of smart algorithms that analyze data and provide us with recommendations, typically for decisions we make on a regular basis, because when we make decisions over and over, the algorithms can learn our preferences and improve the quality of future recommendations (something that isn't quite possible with one-off decisions like: *Which university should I attend*?)

If we extrapolate the consumer-based use of Artificial Intelligence to large organizations, which usually compete in dynamic environments and must deal with the impact of unforeseen events and a multitude of external and internal forces, we can achieve a similar result: namely, improved decision making through intelligent recommendations. In the same way that AI-based apps can improve our decisions for trivial problems (e.g. where to eat, what movie to watch), Artificial Intelligence can also be applied to complex business problems that are difficult to solve through manual methods, and where the consequence of making the wrong decision is much higher than a bad meal or boring movie.

2.1 Decision Making for Complex Business Problems

To understand why and how Artificial Intelligence can improve business decisions, we first need to look at the decision-making process itself. Although different organizations follow different processes for making decisions, they're usually based on the same fundamental steps:

1. Identify the problem (Problem)
2. Gather data on the problem (Data)
3. Organize and interpret data (Information)
4. Understand the "why" (Knowledge)
5. Consider possible solutions, their pros & cons (Evaluation)
6. Implement a solution (Decision)

The above represents a *problem-to-decision workflow*, which is essentially an analytical workflow at its core. This high-level representation of the decision-making process works well for conceptual explanations, but might be the cause of some fundamental misconceptions. An important one, worthy of attention, is the role of knowledge in the process. Most people are familiar with the popular saying, "Knowledge is power!" but what most businesses have come to realize over time is that knowledge by itself won't guarantee the best, or even right decision, even if a business has more knowledge than anyone else. A business may "know" a lot about its customers, but management may still be unsure of what decision to make!

This is because the vast majority of business problems are inherently complex, and thus, difficult to solve. Hence, the decision-making process often breaks down somewhere between the Knowledge and Decision steps, because knowledge in and of itself, isn't quite enough. A closer look at any real-world business problem, whether in distribution, customer retention, or fraud, will bear witness to this obvious truth. Most complex business problems share the following characteristics, which represent the reasons why they're so challenging to solve:

- The number of possible solutions is so large that it precludes a complete search for the best answer
- Real-world business problems are set in dynamic environments
- There are many (possibly conflicting) objectives
- The problem is heavily constrained

Of course, the above list can be extended to include many other characteristics, such as incomplete information (e.g. the necessary data wasn't recorded), noisy data (e.g. the data contain estimates and rounded figures) and uncertainty (e.g. the data aren't reliable). However, these four primary characteristics are sufficient for our purposes, so let's discuss each in turn.

The number of possible solutions is so large that it precludes a complete search for the best answer

Let us assume we want to find the best solution to a problem with 100 decision variables. To keep this example simple, let's also assume that each of these decision variables is binary (i.e. each decision variable can only take one of two possible values, such as "yes" or "no"). Each possible combination of these 100 variables produces some result that can be evaluated and labeled with a *quality measure score,* which is a numerical score that tells us how "good" or "bad" each solution is (similar to a KPI[1] measure). Assume, for example, that a sequence:

"yes" & "yes" & "no" & "no" & "no" & "yes" & "no" & … & "yes"

produces a quality measure score of 79.8, whereas the sequence:

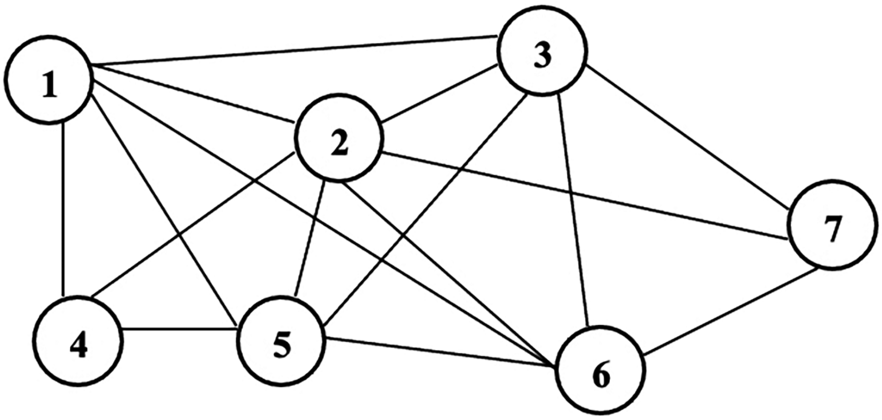
"yes" & "no" & "no" & "yes" & "no" & "yes" & "no" & … & "no"

produces a quality measure score of 91.5. The higher the quality measure score, the better the solution, hence the latter solution is better than the former. Our task is to find the combination of values for the 100 variables that produces the highest possible quality measure score. In other words, we would like to find a solution that cannot be improved.

Without any additional problem-specific knowledge, our approach might be to evaluate all possible combinations. However, the number of possible combinations is enormous. Although each variable can only take one of two values ("yes" or "no"), the number of possible solutions grows at an exponential rate: there are four combinations (2 × 2) for two variables, eight combinations (2 × 2 × 2) for three variables, and so on. With 100 variables, there are 2 × 2 × … × 2 (100 times) combinations—a number that corresponds to 10^{30}. Evaluating all of these combinations is impossible. Even if we had a computer capable of evaluating 1,000 combinations per second, and we began using this computer one billion years ago, we would have evaluated less than 1% of the possible solutions by today!

[1] KPI stands for Key Performance Indicator; it gives a measurable value that demonstrates how effectively a company is achieving key business objectives.

If we revisit the traveling salesman problem introduced in Chapter 1—where traveling the shortest possible distance, the salesman must visit every city in his territory (exactly once) and then return home[2]—recall that with seven cities, the problem has 360 possible solutions,[3] making it relatively easy to solve:



By adding a few more cities, however, the number of possible solutions grows exponentially. To see the maddening growth of these solutions, consider the following:

- A 10-city problem has 181,440 possible solutions
- A 20-city problem has about 10^{16} possible solutions (1 followed by 16 zeros: 10,000,000,000,000,000 possible solutions)
- A 50-city problem has about 10^{62} possible solutions.

By comparison, our planet holds approximately 10^{21} liters of water, so a 50-city problem has more solutions than the number of litres of water on our whole planet! The number of possible solutions to a 100-city problem exceeds by many orders of magnitude the estimated number of atoms in the whole Universe! These numbers are so large they're difficult for us to even conceive of mentally, while most real-world business problems are far more complex than this (in terms of the number of possible solutions). They're defined by a much larger number of variables, and these variables usually take on more values than just "yes" or "no." In such cases, the number of possible solutions is truly astronomical!

[2] Some closely related problems require slightly different criteria, such as finding a tour of the cities that yields the minimum travel time, minimum fuel cost, or a number of other possibilities, but the underlying principle is the same.

[3] For simplicity, we'll assume that the problem is symmetric (i.e. the distance between cities *A* and *B* is the same as the distance between *B* and *C*). Note also, that solution 1–2–3–4–5–6–7 is the same as solution 3–4–5–6–7–1–2, as both these solutions have a different starting city but represent the same cycle.

So, how can we find optimal solutions to such problems? An exhaustive search that relies on computing power is clearly not the answer, as the number of possible routes, fraud rules, or transportation plans might be so large that examining all possibilities—with even the fastest supercomputers—would take many centuries at best. In the following chapters, we'll explore a real-world business problem where the number of possible solutions is *much* larger than the numbers presented here and show how such problems can be solved using Artificial Intelligence methods.

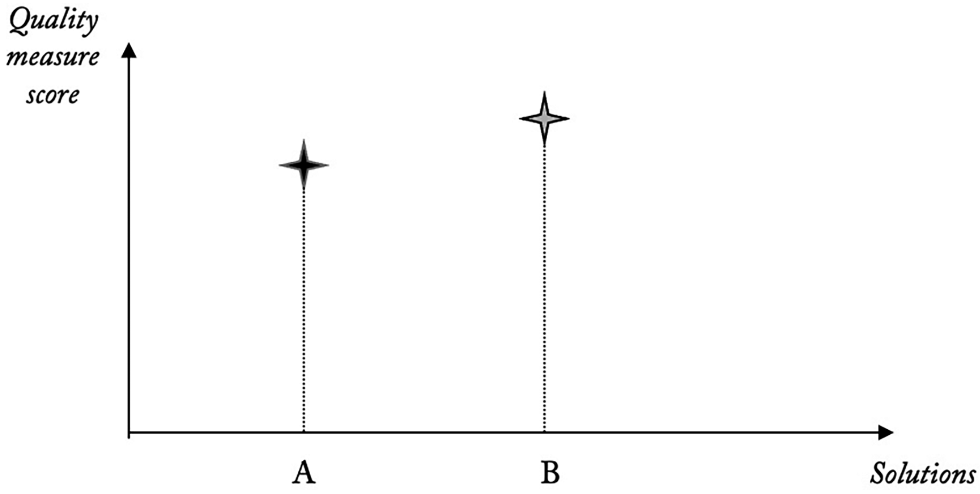
The problem exists in a dynamic environment

Business managers know that real world problems aren't static, and yet they take static snapshots of the problems they're trying to solve. Such snapshots represent a good starting point for analyzing and understanding a problem, but on their own, they paint a false picture. Because real-world problems are set in dynamic, time-changing environments, we must address the time factor explicitly. To illustrate this point, let's consider a real-world version of the traveling salesman problem with delivery trucks. If the problem is carefully analyzed and a set of delivery routes found, the quality of these routes will be affected by many factors, such as rush-hour and weekend traffic, weather and road conditions, and so forth, as well as random events, such as labor strikes or accidents. Because the problem is influenced by so many external factors, any solution to a static snapshot of this problem might prove inadequate.

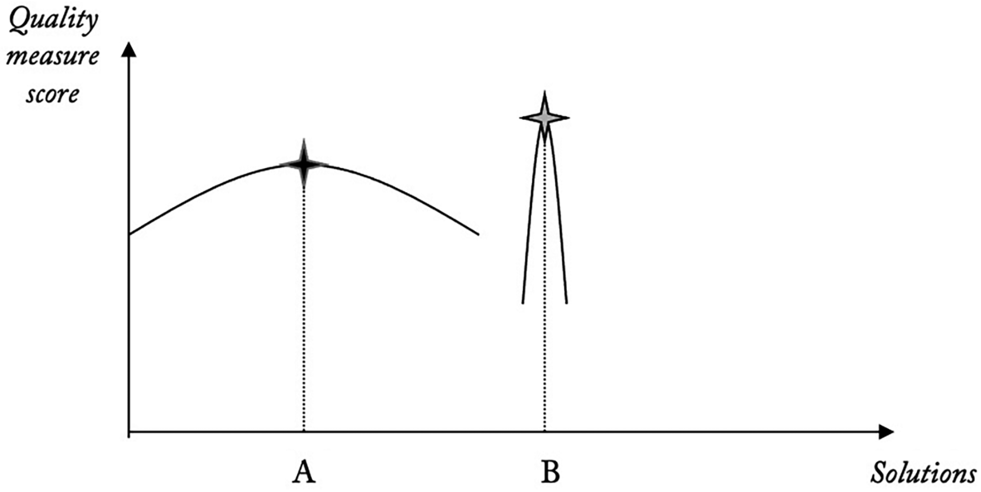
We can take another example from sales operations, where significant effort is taken each year to optimize a sales rep's territory and determine the optimum number of visits that each customer should receive within each call cycle. This static approach to the problem is bound to result in the under- and over-servicing of customers during the course of a year, because some customers will change in volume and importance, but the static solution won't consider these changes. Hence, as time passes, the static solution will deteriorate in quality (becoming more sub-optimal) to the point where sales resources could have been put to far better use by pursuing new opportunities or proactively managing customer churn.

This *dynamic environment factor* becomes even more prevalent in manufacturing and distributed supply chains, where static plans and schedules may face steep deterioration in performance when exposed to dynamic variables such as process variability, equipment failure, weather events, and demand spikes, versus more "robust" and "forgiving" plans, which are more tolerant of unexpected events, changes, and modifications.

There are some additional issues related to dynamic environments that are worth noting. Imagine that we are considering two solutions, A or B:



Which of these two solutions should we select? Well, the question seems trivial: Because solution B has a higher quality measure score, solution B is better than solution A. Although this statement is true—solution B *is* better than solution A—the answer might not be that straightforward. It may be the case that solution A "sits" on a relatively flat peak, whereas solution B "sits" on a very narrow peak:



We can interpret the above graph as follows: Although Solution B is better than solution A (there is no doubt about that), it is *peak optimized*—meaning that nothing can go wrong for us to achieve the result (all our assumptions must pan out, no exceptions!). Hence, if we're forced to modify solution B in any way (due to process variability, unexpected maintenance, demand spikes, or some other reason), then the quality of Solution B will deteriorate very quickly. Solution A, on the other hand, has a lower quality measure score to begin with, but achieving this result is far more likely because the solution can tolerate changes and modifications without a sharp drop-off in quality. Given that solution A is less risky than B, should we still select the "better" Solution B?

There are many (possibly conflicting) objectives

It's quite unusual for any real-world business problem to have only one objective. In fact, in recurring cost-cutting environments the go-to position always seems to involve the conflicting request to "do more with less!" Complex problems, especially where there's a lot at stake, often involve a range of objectives that could be working against one another. Such problems are called *multi-objective problems*, as there is more than one objective to satisfy, and an increase in the quality measure score for one objective might come at the cost of another.

A simple example of this phenomenon exists in manufacturing, where companies try to carry just enough inventory to satisfy future customer demand, without carrying too much. By keeping inventory levels high, a manufacturer can be sure to satisfy future customer demand along with any unexpected spike in orders, but this approach can have significant working capital implications, as well as potential obsolescence costs (especially in sectors such as food and electronics, where inventory is either perishable or quickly becomes obsolete). By keeping inventory levels low, on the other hand, the manufacturer can improve its operational metrics (such as stock turns) and realize substantial savings in working capital and obsolescence costs, but is likely to experience occasional stock-outs and lost sales. Hence, there's a trade-off between the competing objectives of minimizing inventory costs and maximizing customer service levels.

In multi-objective problems, maximizing the performance of one objective (such as cost) is likely to come at the expense of other objectives (such as safety, time, or service levels), thereby rendering the concept of a single "best solution" no longer relevant. Instead of a single optimal solution, such problems have many optimal solutions, with each solution performing better or worse against the selected objectives, thereby leaving the decision-maker with the complex task of evaluating these trade-offs.

The problem is heavily constrained

All real-world business problems have constraints of some sort, and for a particular solution to be suitable for consideration, it should satisfy many restrictions imposed by business rules, capacities, contractual obligations, regulations, laws, and/or preferences.

For example, let's consider the problem faced by Australian pharmaceutical wholesalers, which distribute medicines that carry *Community Service Obligation* requirements. Part of these service obligations require the supply of a full set of medicines to pharmacies across Australia usually within 24 hours, regardless of location and cost of supply! Now consider the number of constraints involved in coming up with a delivery plan:

- The number of delivery vehicles and their location (e.g. more than one-quarter of all Australian pharmacies are located more than 100 km from the nearest capital city)
- The desired delivery time window of each pharmacy
- Orders needing to be delivered in less than 24 hours
- Certain medicines are temperature sensitive and require special storage or specialized vehicles

It's also important to note that some of these constraints are mandatory (referred to as *hard constraints*, such as the number of delivery vehicles), while others may be flexible (referred to as *soft constraints*, such as delivery time windows).

Pharmaceutical wholesalers employ teams of people to solve such problems in a way that creates the best outcome for all parties—but what does "best" mean? Well, in this case, it might mean a plan that satisfies all constraints and has the lowest overall cost of implementation (i.e. a plan that is within the total funding provided by the government and is able to meet the contractual service levels). The challenge, however, is that sometimes finding even one plan that satisfies *all* constraints can be quite difficult.

Decision-making process for complex business problems

Consistent, high-quality decisions in any industry can be traced back to the effectiveness of the problem-to-decision workflow discussed earlier:

Problem > *Data* > *Information* > *Knowledge* > *Evaluation* > *Decision*

This workflow has traditionally been implemented in a manual way in most organizations, through the use of human experts and analysts. For this reason, the extent to which this workflow effectively bridges the gap between "knowledge" and "good decisions" depends on the nature and complexity of the problem, as well as the amount and quality of resources applied to the "evaluation" step.

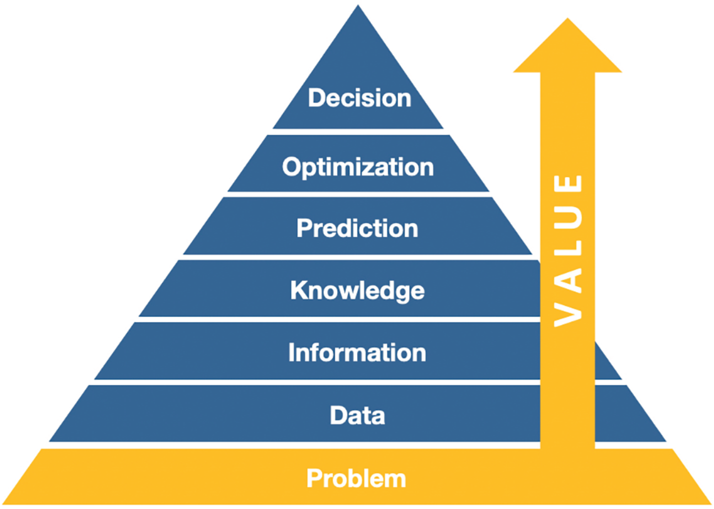
For simple problems with few possible solutions, no conflicting objectives, and minimal constraints, the evaluation step can be managed through manual methods. However, as the number of possible solutions grows, as the influence of dynamic variables increases, as multiple (competing) objectives are introduced, and as more and more constraints and business rules are applied, the problem grows exponentially in complexity and the evaluation step becomes more difficult—perhaps even impossible—to undertake through manual efforts alone. As an example, if a complex problem has millions of possible solutions, with many trade-offs among objectives, the time it would take to find and evaluate all these solutions would be prohibitive (i.e. centuries).

This means that the more complex the problem (i.e. the greater the number of possible solutions, dynamic variables, conflicting objectives, and constraints), the more difficult the evaluation step, and throwing more resources at the problem is unlikely to improve the decision for the simple reason that it's difficult just to identify all possible solutions, to say nothing of evaluating them in detail. This puts a ceiling on the complexity of problems that an organization can effectively address through manual efforts, and raises the question of whether we can automate the problem-to-decision workflow for recurring decisions? And if so, how?

To answer these questions, let's look at the various levels of sophistication (and related approaches) that are available to any organization when it comes to decision making, and then discuss the role that Artificial Intelligence can play.

2.2 The Problem-to-Decision Pyramid

The diagram below represents the continuum that exists in terms of an organization's ability to improve its decision making. The best way to understand this diagram is through an analogy of climbing a pyramid, where the higher we climb, the further we can see. Using this analogy, each layer represents a step in the quest for improved decision making, so the higher we go, the better our decisions.

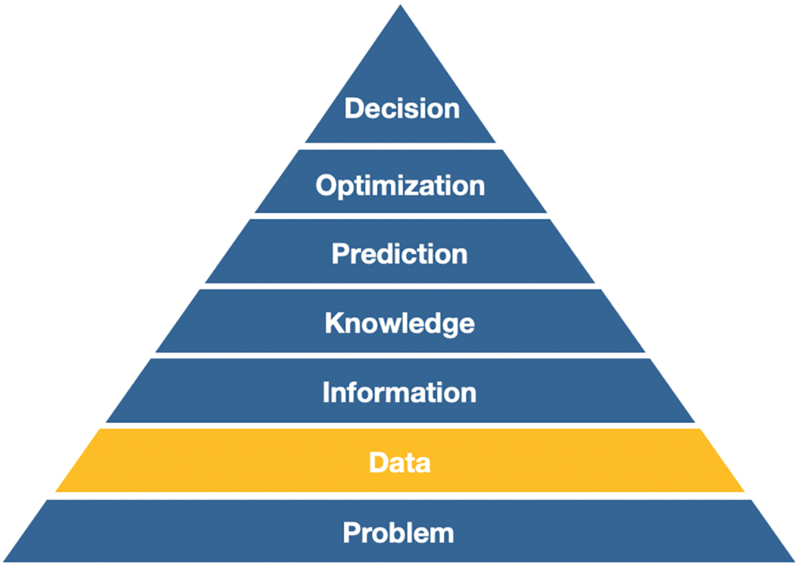


Problem

The *problem* layer is the foundation of the pyramid and represents the specific business problem we're trying to solve. No matter how big or small the problem, this represents the first step in the decision-making workflow, and in many ways, the most important, because we can't climb the pyramid without first identifying and understanding the business problem. For example, we can't collect the necessary data (the next layer of the pyramid) without first defining the problem.

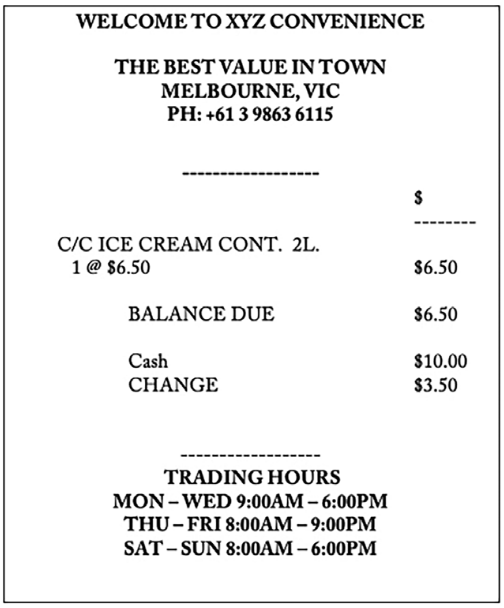
Data

The second step in our pyramid involves the collection and storage of data pertaining to the problem we're solving:



The word *data* means "known facts." As a general concept, it refers to the situation where some existing facts (whether qualitative or quantitative) are represented or coded in a form suitable for usage or processing. Data are collected in the form of bits, numbers, symbols, and objects, and a typical piece of data consists of a pair (attribute, value), such as "color, red." Data can be pre-processed, cleaned, arranged into structures, stripped of redundancy, and organized or aggregated to provide *information*, which is the next layer of the pyramid.

To have a better appreciation of what data looks like, let's have a look at the different *attributes* and *values* in the receipt below, where the "$" sign stands for the attribute "sales price," and "L" stands for the attribute "container volume (in liters)." In the same receipt, we can see a few examples of values, such as "$6.50" for the price, "2L" for the container volume and "1" for the number of units being purchased:



WELCOME TO XYZ CONVENIENCE

THE BEST VALUE IN TOWN
MELBOURNE, VIC
PH: +61 3 9863 6115

```
                 ------------------
                                              $
                                          --------
C/C ICE CREAM CONT.  2L.
  1 @ $6.50                                 $6.50

        BALANCE DUE                         $6.50

        Cash                                $10.00
        CHANGE                              $3.50

                 ------------------
```

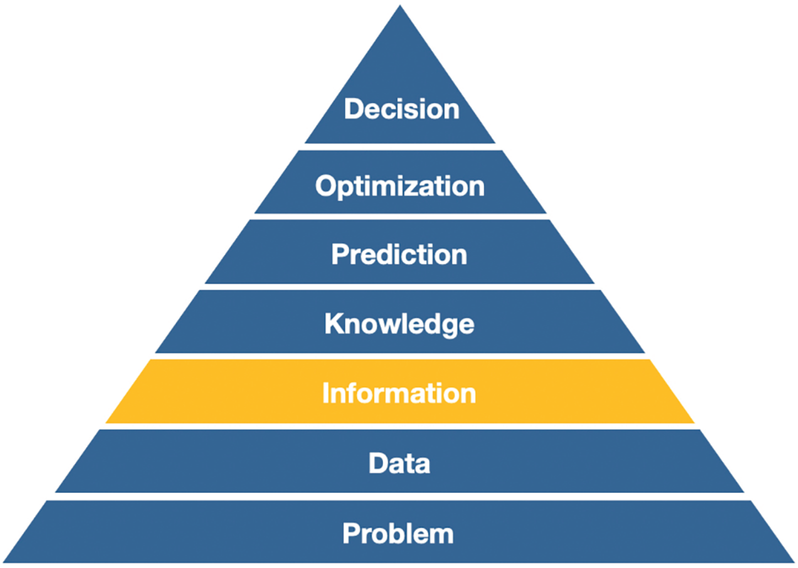
TRADING HOURS
MON – WED 9:00AM – 6:00PM
THU – FRI 8:00AM – 9:00PM
SAT – SUN 8:00AM – 6:00PM

Recent years have seen a growing obsession with data: collecting it, mastering it, reporting on it, and in some cases, even valuing it like a financial asset. That is understandable: after all, data is the first step in the problem-to-decision workflow. However, many organizations—usually well-funded and well-resourced enterprises—have been collecting and storing large volumes of data for years on the premise that "if we collect good data, then good decisions will follow," only to discover that the connection between data and decision making isn't automatic because of the other steps in the problem-to-decision workflow. Hence, we must make a distinction between "good data" (second layer of the pyramid) and "good decisions" (top of the pyramid), and a further distinction between "good" and "bad" data—after all, what is *good* data anyway? There is some temptation to answer this question in terms of the state of the data (i.e. quantity, quality, timeliness, structure, etc.), but in the context of improving business decisions, "good" refers to any data that assists us in diagnosing, explaining, and assessing the problem we're trying to solve.

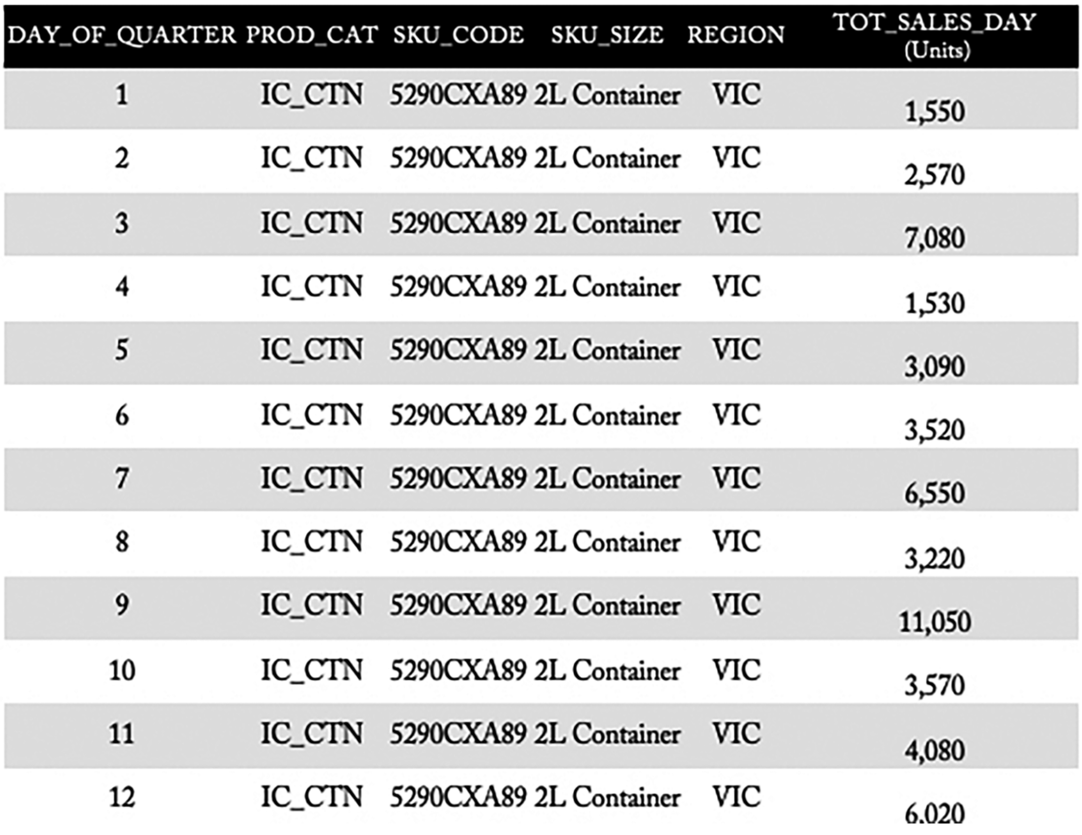
From this perspective, we should look beyond the boundaries of the organization and consider external data as well, such as demographics, weather, point-of-sale transactions, competitor pricing, government approvals and licenses, and so on. As an example, census data can be used to understand the demographical characteristics of customer groups across various geographic areas, which in turn can be used to better understand why certain product promotions are more effective in some areas than others.

Information

The next layer of the pyramid, *information*, includes facts and relationships that have been perceived, discovered, or learned from the data.

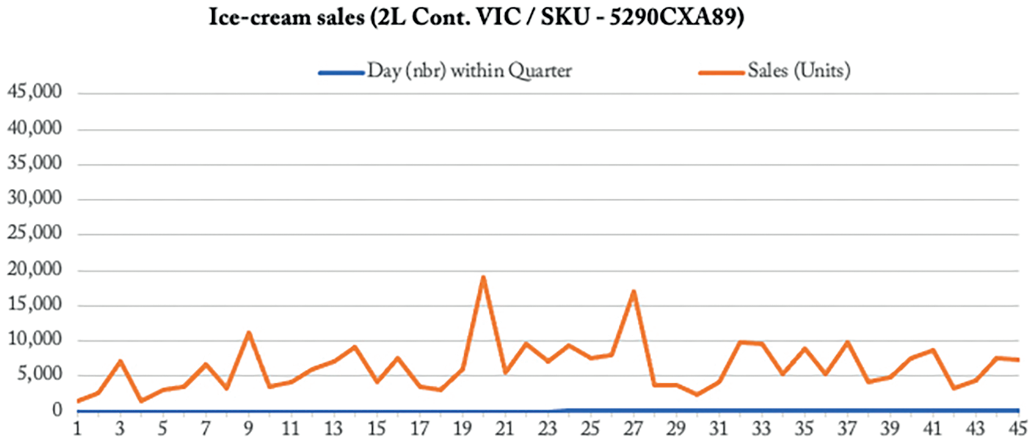


The *information* layer of the pyramid leverages reporting and data visualization techniques to graphically represent data, with the output being reports, charts, graphs, statistical tables, and more. As an example, the supermarket receipt above contains data on ice-cream sales, which the manufacturer could aggregate into an informational report to better understand sales performance in a specific region (in this case, Victoria):



DAY_OF_QUARTER	PROD_CAT	SKU_CODE	SKU_SIZE	REGION	TOT_SALES_DAY (Units)
1	IC_CTN	5290CXA89	2L Container	VIC	1,550
2	IC_CTN	5290CXA89	2L Container	VIC	2,570
3	IC_CTN	5290CXA89	2L Container	VIC	7,080
4	IC_CTN	5290CXA89	2L Container	VIC	1,530
5	IC_CTN	5290CXA89	2L Container	VIC	3,090
6	IC_CTN	5290CXA89	2L Container	VIC	3,520
7	IC_CTN	5290CXA89	2L Container	VIC	6,550
8	IC_CTN	5290CXA89	2L Container	VIC	3,220
9	IC_CTN	5290CXA89	2L Container	VIC	11,050
10	IC_CTN	5290CXA89	2L Container	VIC	3,570
11	IC_CTN	5290CXA89	2L Container	VIC	4,080
12	IC_CTN	5290CXA89	2L Container	VIC	6.020

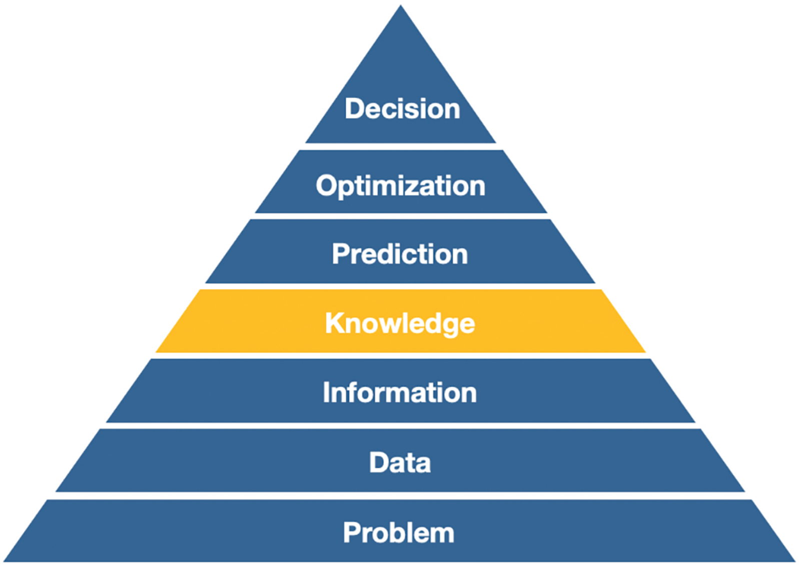
The same information can also be visualized in a chart:



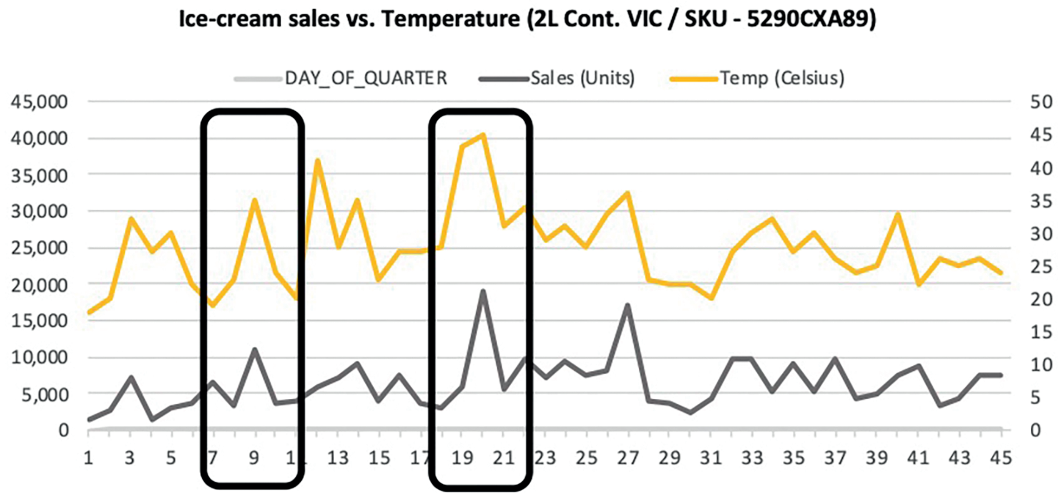
Effective outputs of this layer of the pyramid can communicate complex information clearly and efficiently, making it easier for humans to understand trends, outliers, and patterns. While a critical step in the problem-to-decision workflow, this layer of the pyramid represents the most basic level of analytics and is usually referred to as *descriptive analytics*. When done well, it might suffice as a decision support tool for smaller-scale, simpler problems, but would be inadequate for business problems of greater scale and complexity, as discussed above.

Knowledge

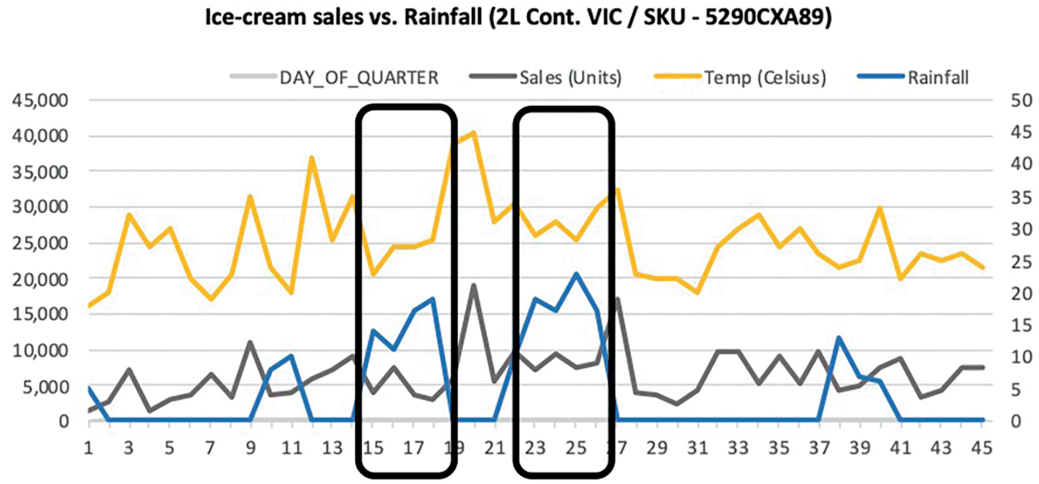
If the *information* layer of the pyramid tells us *what* happened, then the *knowledge* layer tells us *why* it happened. This layer of the pyramid builds on the outputs from the previous layers to provide a deeper understanding of both the data and the problem we're trying to solve. Unsurprisingly, many refer to this layer as *diagnostic analytics*:



The goal of the *knowledge* layer is to develop a good understanding of what happened in the past, the factors (i.e. variables) that contributed, relationships between those factors (i.e. correlations), and possibly the extent to which any single variable contributed to the result more than any other (dominant variable). In the ice-cream sales example above, the chart shows us what happened (i.e. information on how many units were sold), but in the *knowledge* layer of the pyramid, we would like to understand *why* it happened by establishing a correlation between consumer demand for ice-cream and other variables, such as changes in price or temperature. This knowledge could then be communicated in a number of ways, such as the chart below, plotting sales units alongside changes in temperature:



To expand our knowledge, we may want to explore if rainfall has any further effect on consumer demand, and create a complete graph exploring the movement of sales alongside changes in local temperature and rainfall:

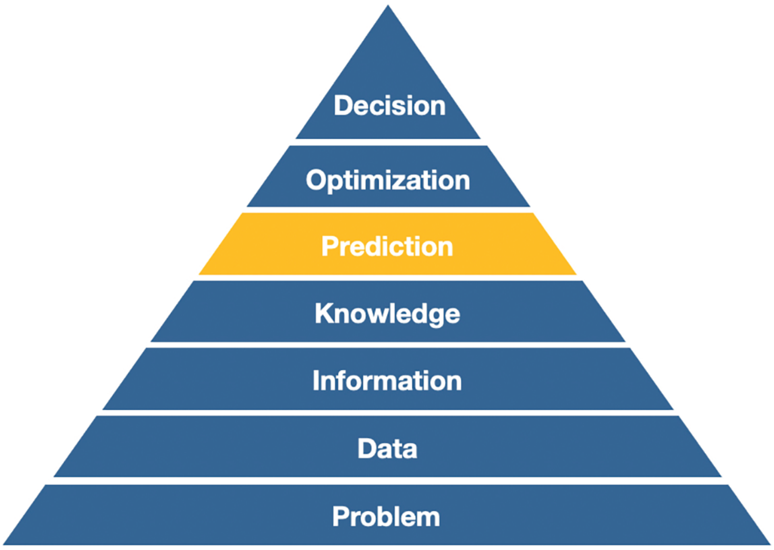


This process can then be repeated, where we search for possible relationships and correlations to other variables, such as competitor pricing or promotional activity, among others.

Prediction

The next layer of the pyramid deals with answering the question: *What will happen next*? So if the *information* layer tells us *what happened* and the knowledge layer tells us *why it happened,* then the prediction layer tells us *what might happen in the future* (with some probability) and is often referred to, unsurprisingly, as *predictive analytics*. Hence, in the problem-to-decision workflow:

Problem > *Data* > *Information* > *Knowledge* > *Evaluation* > *Decision* the evaluation step is now expanded to include *prediction* and *optimization,* which are essential for identifying possible solutions, predicting their outcome, and assisting in their evaluation.



A key feature of the *prediction* layer of the pyramid is the ability to predict outcomes for various scenarios that can be interpreted as "what-if" questions. For example, the question might be: *What is the impact on customer services levels, if one of the following happens* (the "what" and the "if"):

- If we buy three additional delivery trucks?
- If we change the overnight location of the delivery trucks?
- If we build another distribution center in Victoria?
- If we have to service 10% more customers?
- If we have to carry 5% more products?

Continuing with the ice-cream sales example from above, let's say the manufacturer is preparing to launch a new product in Australia and needs to identify a territory that meets a specific set of qualification criteria for the launch. For example, the area must have a high volume of existing ice-cream sales, an older and affluent demographic, and a minimum number of retail outlets. To further complicate the search for the right territory, the new product must meet a specific sales target for the month of its launch, and given the size of Australia and the timing, weather is likely to play a role (as well as the advertising budget).

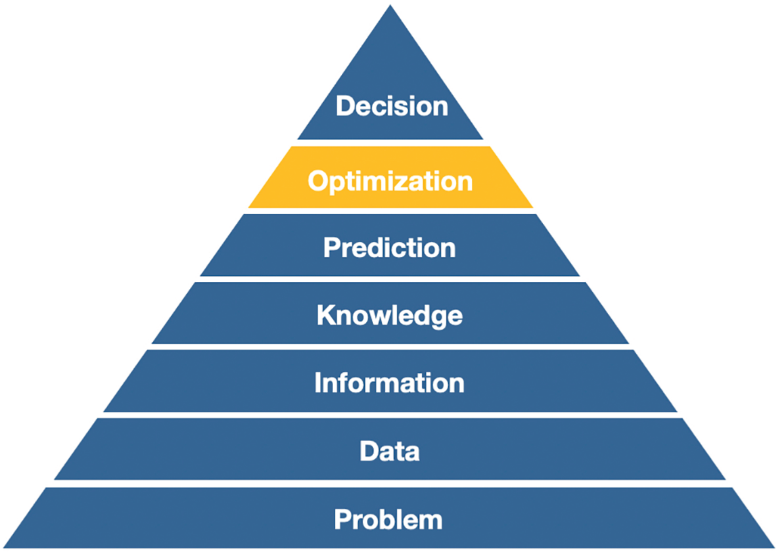
To make a decision for the launch, the manufacturer would need to consider several scenarios (i.e. what-if questions) to find the best territory. Each what-if question, (e.g. *What will sales be if the new product is launched in Victoria during the month of April?)* requires a few core elements, which together constitute a single scenario:

- *Data for each territory:* The number (and characteristics) of retail outlets, seasonal weather patterns, demographics of each catchment area, historical sales volumes for relevant products, field sales staff and territory structures, as well as other pieces of data that might affect the predicted outcome (in this case, sales of the new product).
- *Constraints that define a possible solution:* These could be: (1) the minimum sales target that must be achieved for the new product; (2) that sales of the new product must not cannibalize sales of existing products; (3) the characteristics of the selected territory must be representative of the broader target market (e.g. high volume of ice-cream sales for specific product ranges and an older and more affluent consumer base); and so on.
- *Objective:* The specific metric for which we are predicting the outcome, in this case, finding the territory and month that satisfies the qualification criteria and generates the highest sales for the new product.

While this layer of the pyramid can enable quite sophisticated capabilities (i.e. predicting outcomes), we are still faced with a substantial limitation. Recall that in complex business problems, there is an extremely high number of possible solutions (i.e. scenarios to investigate), and more often than not, we are working with multiple objectives simultaneously. Given the complexity of such an iterative what-if planning process, we'll only have time to create and evaluate a limited number of scenarios. Consequently, the chances of finding the best solution are rather slim. Which raises the question: If we had time to create and evaluate millions of scenarios, could we find better a solution—one that satisfies all problem-specific constraints and has an overall higher level of predicted sales? The answer is yes, which moves us to the next layer of the pyramid, *optimization*.

Optimization

If the *prediction* layer tries to answer the question of *what will happen* in the future for any given scenario, then the *optimization* layer tries to identify the scenario that provides the "best" outcome while satisfying all constraints and business rules—in other words, the best solution to the problem we're trying to solve.



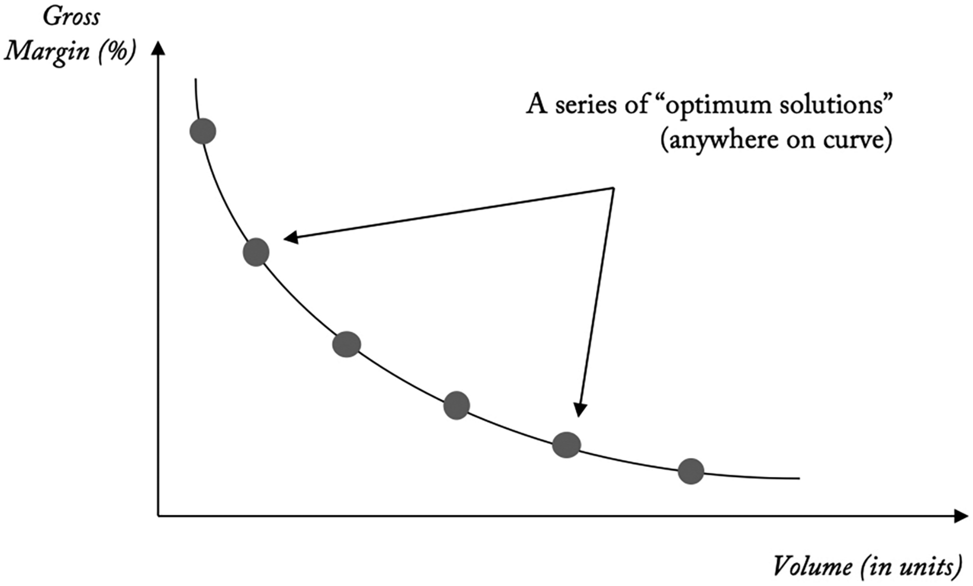
If we go back to the ice-cream sales example, the manufacturer might want to grow market share through the use of promotions. But to make the best possible decision around what promotion to use for what product in what territory (and when), the manufacturer would need to evaluate a huge number of possibilities (scenarios), with each scenario being a promotional plan that can be executed in the marketplace. Given the astronomical number of such scenarios, where each one represents a unique promotional plan that requires evaluation, this workflow is quite intricate and involved, especially that it needs to take into account all key variables, such as:

- *Products*: Do certain products respond better to promotions than others?
- *Promotion type:* What type of promotion (e.g. individual discounts, two-for-one offers, multi-buy discounts) should be applied to each product?
- *When*: What day? week? month?
- *Where*: What territories? retail chains? stores?
- *Duration:* Weekend only? entire week? two weeks?
- *Regularity:* Will the promotion be one-off? or repetitive? If repetitive, what should be the gap, if any, between promotions (e.g. 2 weeks on followed by 2 weeks off)?

Aside from all the possible combinations of different values of the above variables, the manufacturer must also consider various business rules and constraints. As a simple example, some retailers might impose restrictions on the frequency/regularity of promotions for a given product, while others might restrict the entire ice-cream category to only a few types of promotions.

To complicate matters further, the "best" solution might need to consider multiple objectives and their trade-offs. To grow market share, the manufacturer will need to maximize sales volume, but this objective competes against another major objective: margin and profitability—meaning that sales volume can be maximized by increasing the promotional discount, but such discounts drive down the manufacturer's margin (and perhaps overall profitability). On top of this, the retailer's margin needs to be considered, because the proposed promotion will be rejected if it doesn't achieve the retailer's margin and profit objectives, which in turn trade-off against the manufacturer's margin and profit objectives (and so on). These are only a few examples of conflicting objectives, where improvements on one objective come at the expense of another.

Once multiple solutions are created, evaluating the quality of each promotional plan becomes a complicated task, as it requires predicting the outcome of each plan while considering the impact from other variables (e.g. *Does this promotional plan decrease the sales volume of other products? And if so, by how much?*), as well as determining if any constraints or business rules have been violated (which might mean that a particular plan isn't a possible solution after all). Once the predicted sales volume for several promotional plans has been generated, the corresponding plans can be plotted on a graph to visualize their performance and investigate trade-offs:



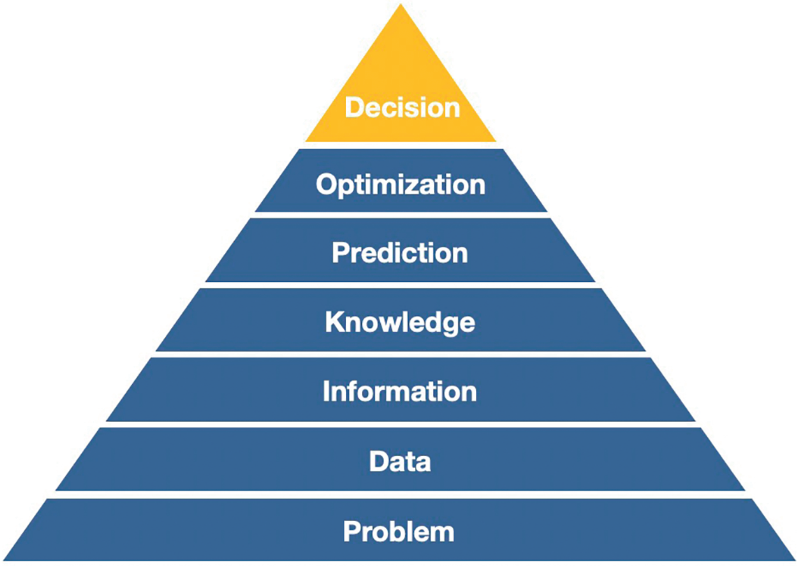
The above curve is termed a *Pareto front*, which takes its name from the famous Italian economist. A solution is *Pareto optimal* if it's impossible to improve any of the objectives without decreasing at least one other objective—therefore, all solutions on the Pareto front are Pareto optimal, and any solution that's not on the curve isn't optimal. In this case, the manufacturer is provided with a set of Pareto optimal promotional plans that show the trade-off between gross margin and unit volume.

Complex business problems like this are not well-suited for manual and spreadsheet-centric approaches for a few reasons: first, given the complexity of evaluating each scenario, manual methods can lead to biased, error-prone, and inaccurate predictions, as they're often based on gut feel and intuition. Second, it's impossible to manually create an extensive set of scenarios that cover a large number of possible combinations of key variable values, as decision-making timeframes don't allow for that. Without a sophisticated tool or system, creating these scenarios, evaluating them, and then analyzing the various trade-offs becomes an impossible task.

To see optimization in "action," and learn more about multi-objective optimization, we encourage you to watch the supplementary video for this chapter at: www.Complexica.com/RiseofAI/Chapter2.

Decision

Climbing past optimization brings us to the *decision* layer, which is the capstone of the problem-to-decision pyramid.

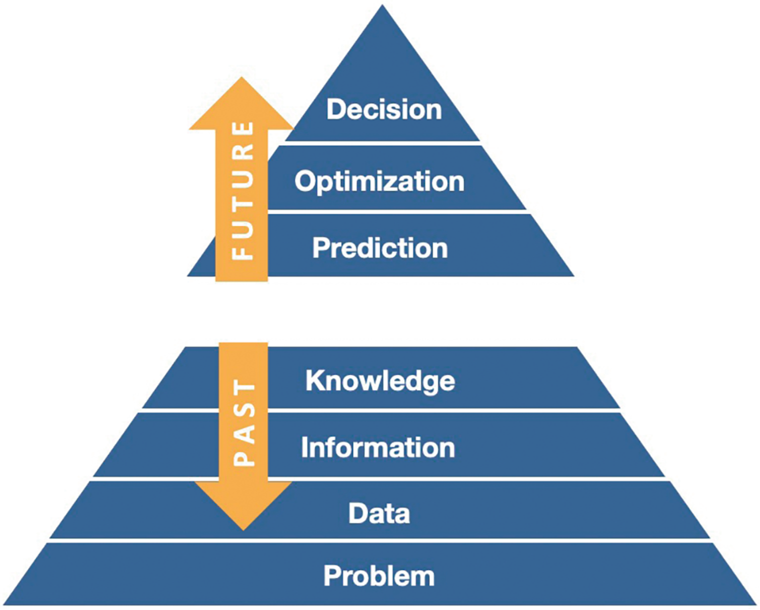


The *decision* layer represents the actual data-driven, optimized decision that is made, and which is enabled through the capabilities at each layer of the pyramid. In the ice-cream sales example, where the manufacturer wanted to grow market share through the use of promotions, the final decision would take the form of an optimized promotional plan. All KPIs, constraints, business rules, and trade-offs would have to be taken into account, and a handful of solutions (i.e. possible promotional plans) would be presented on a trade-off graph, so that the manufacturer can make a well-informed decision. The final plan that's selected and implemented would combine the right mix of products, types of promotions, and timeframes to deliver results against the conflicting objectives of maximizing gross margin and unit volume. Also, once the manufacturer's decision has been implemented and the results are known, the outcome needs to be fed back into the decision-making workflow so that the manufacturer can make even better decisions in the future (in effect "learning" from the outcome of previous decisions).

This problem-to-decision pyramid presents a compelling climb for most organizations, and it's easy to understand why. After all, being able to consistently make data-driven, optimized decisions can unlock significant value in most organizations, as discussed in Chapter 1.4. On the flipside, not making the climb—or attempting the climb with a labor-intensive and spreadsheet-centric approach—is likely to facilitate poor decisions that destroy value and allow competitors to gain an advantage. With that in mind, let's now explore the benefits that Artificial Intelligence methods can bring to the problem-to-decision pyramid.

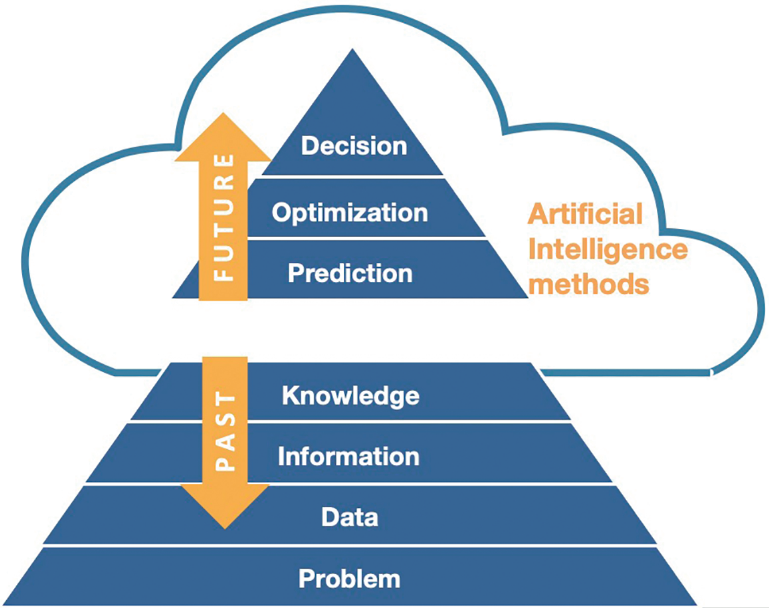
2.3 AI for Bridging the Gap between Past & Future

Organizations wishing to climb and progress through this problem-to-decision pyramid must realize that a gap exists in the climb, requiring a step-change in capability to bridge and enable further progress. The reason for this gap is because the lower layers in the pyramid deal with the past (reporting on it, explaining why certain things happened), while the top layers deal with the future (predicting it, finding optimized solutions).



Bridging this gap between the past and future isn't straightforward, because the sophistication required for predicting the future and optimizing decisions is far greater than that of reporting on the past. One example of this difference lies in the fact that past patterns might not continue into the future, which isn't something we need to worry about if we're only reporting on the past, but something we need to deeply consider when trying to predict the future (requiring a greater level of sophistication). For this reason, organizations that have climbed to higher levels of the pyramid usually make better decisions than organizations that haven't.

When it comes to making predictions and recommending an optimized course of action, there are many tools and enterprise software applications available that can help organizations reach the top of the pyramid, and which are usually based on some sort of Artificial Intelligence method. The point is that as an organization climbs up the pyramid, the sophistication of the required tools and technologies also increases, with Artificial Intelligence having the most applicability and delivering the most value within the *prediction* and *optimization* layers of the pyramid, mainly because that's where the greatest complexity resides.



Keeping this in mind, we can use various Artificial Intelligence methods to develop a system capable of recommending optimized decisions, as well as learning from previous actions and decisions. Building such a *Decision Optimization System* involves three fundamental steps:

1. Building a predictive model
2. Building an optimization model
3. Incorporating adaptability (feedback loop)

These steps are briefly discussed below, and then again in Part II of this book, which explores these subjects in far more technical detail.

Building a predictive model

The first step in bridging the gap between past and future is developing a capability to predict what will happen next. To enable such a capability, we first need to identify the relationships and patterns among the various variables in the data, and then use our understanding of these relationships and patterns to build and train a model (or set of models) capable of predicting some outcome. This process is explained in greater technical detail in Chapter 5, but for now, it's only important to understand that the accuracy of the prediction is directly related to the quality and granularity of the underlying model. If the model has too many vague assumptions and approximations, the prediction may be meaningless, or worse.

For these reasons and others, creating such models requires an iterative, flexible, and cyclical approach, involving a set of tasks, usually referred to as a *Data Science Methodology.* One of the most widely used methodologies, the *CRISP-DM's methodology* (Cross-Industry Standard Process for Data Mining) includes the following steps:

- Business understanding
- Data understanding
- Data preparation
- Modeling
- Modeling evaluation
- Modeling deployment

Building an optimization model

The next step is to build an optimization model (covered in more detail in Chapter 6), which requires us to define:

- *Variables and their domains*: For example, a variable in a promotional plan might be the type of promotion (e.g. in-store, catalogue, etc.), and its domain would be the set of possible values: (10% off, 20% off, 25% off, etc.).
- *Constraints and business rules that define a feasible solution*: For example, the min/max frequency (of promotion for a particular product).
- *Objective*: For example, the total volume sold.

Clearly, there are additional details that need to be specified—these include categorization of constraints/business rules into soft and hard (Chapter 3 provides a more detailed discussion on this topic), possible penalties for violation of soft constraints, the relative importance of different KPIs, among others.

It's also important to note that the optimization model will work closely with the predictive model in the following way:

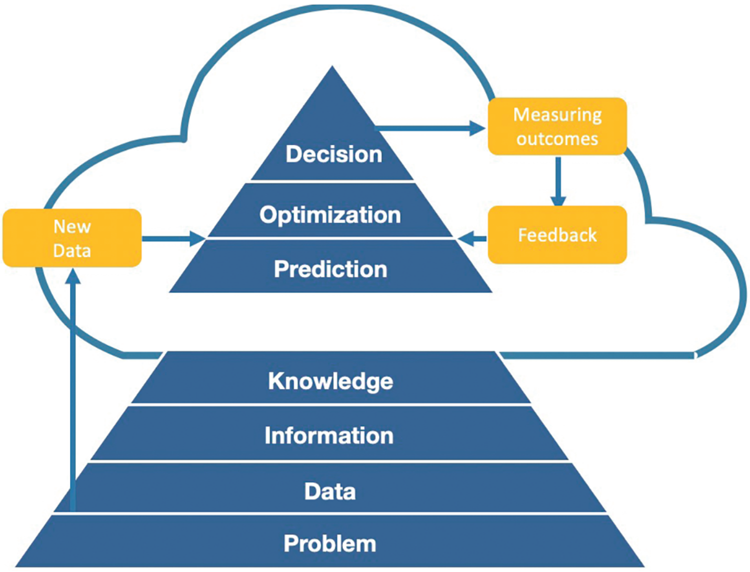
- The *optimization model* will automatically generate many possible (future) scenarios.
- The *predictive model* will evaluate each scenario generated by the optimization model against a single or multiple objectives and constraints, and generate a predicted outcome that is sent back to the optimization model for further action.

This approach allows the two models to "talk" to one another, and find possible solutions that satisfy all problem-specific constraints and business rules. The building, training, and deployment of such models is a significant technical undertaking that involves a great deal of technical expertise from AI scientists that specialize in various algorithmic methods.

And lastly, to fully bridge the gap between the past and future, these two models need to be augmented by the third step: the introduction of *adaptability*. Recall that complex business problems exist in dynamic environments and yet the predictive model is based on a static snapshot of historical data!

Incorporating adaptability (feedback loop)

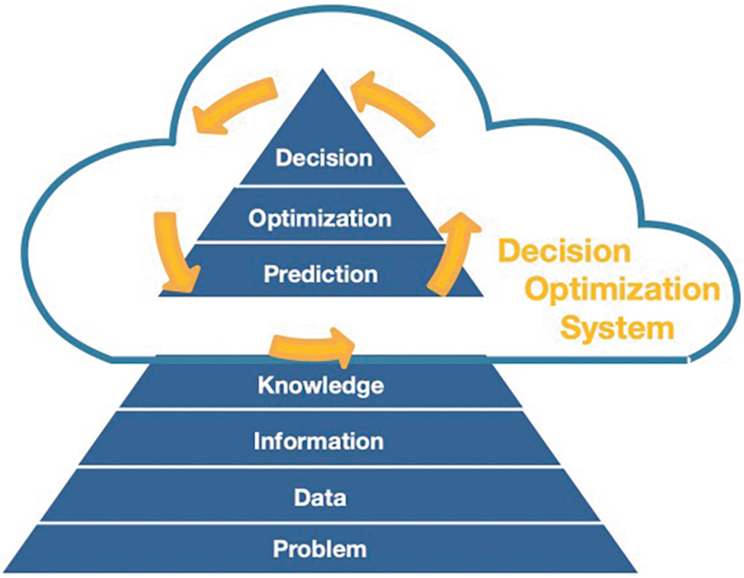
The third step leverages the arrival of new data and feedback on current performance so that the models can learn from the outcome of past decisions, in order to make more accurate predictions and recommend better decisions in the future. More specifically, the models need a mechanism for "knowing" what actually happened (versus what was predicted to happen), and for updating themselves accordingly by taking these "actuals" into account when making future predictions or recommendations.



This feedback loop is a critical component of any Decision Optimization System and essential for ongoing optimized decision making in any dynamic environment. If the underlying models remain static, the system would lose accuracy and relevance, and eventually grow obsolete—in some cases, very quickly. By incorporating a mechanism for self-learning, the system can monitor outcome data and calculate the variance between predicted and actual values. When they vary beyond specific thresholds, it will trigger the system's adaptive algorithms to update the underlying models (automatic self-tuning—for example, by creating new (emerging) rules. Chapter 7 explains how this works.

Recommending the best decision

Once the underlying models have been trained and deployed, and the feedback loop enabled, the Decision Optimization System is ready to automate the problem-to-decision workflow and provide optimized recommendations.



As discussed earlier, business problems grow in complexity as the number of possible solutions grows, as the influence of dynamic variables increases, as multiple (competing) objectives are introduced, and as more and more constraints and business rules are applied. And as these problems grow in complexity and the evaluation step becomes more difficult (and sometimes impossible), they become increasingly difficult to address through manual efforts. Hence, the benefit an organization can gain through a Decision Optimization System based on Artificial Intelligence methods lies in faster, and consistently higher-quality decisions, which will impact key metrics such as margin, sales growth, market share, production costs, and more.

To explain how Artificial Intelligence and Decision Optimization Systems work on real-world problems, we'll delve into a well-known and highly complex business problem in the next chapter, which involves manufacturers and retailers and how they struggle to maximize revenue, margin, and growth. Also, for more information about the problem-to-decision pyramid, along with examples of AI-based systems for prediction and optimization, please visit the supplementary video for Chapter 2 at: www.Complexica.com/RiseofAI/Chapter2.

Index