Data Analytics for IT Networks

Developing Innovative Use Cases

John Garrett
CCIE No. 6204 Emeritus
MS Predictive Analytics

ciscopress.com

FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Data Analytics for IT Networks

Developing Innovative Use Cases

John Garrett CCIE Emeritus No. 6204, MSPA

Cisco Press

Data Analytics for IT Networks

Developing Innovative Use Cases

Copyright © 2019 Cisco Systems, Inc.

Published by:

Cisco Press

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

First Printing 1 18

Library of Congress Control Number: 2018949183

ISBN-13: 978-1-58714-513-1
ISBN-10: 1-58714-513-8

Warning and Disclaimer

This book is designed to provide information about Developing Analytics use cases. It is intended to be a guideline for the networking professional, written by a networking professional, toward understanding Data Science and Analytics as it applies to the networking domain. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

MICROSOFTAND/OR ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THE INFORMATION CONTAINED IN THE DOCUMENTS AND RELATED GRAPHICS PUBLISHED AS PART OF THE SERVICES FOR ANY PURPOSE. ALL SUCH DOCUMENTS AND RELATED GRAPHICS ARE PROVIDED "AS IS"

WITHOUT WARRANTY OF ANY KIND. MICROSOFT AND/OR ITS RESPECTIVE SUPPLIERS HEREBY DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS INFORMATION, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL MICROSOFT AND/OR ITS RESPECTIVE

SUPPLIERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF INFORMATION AVAILABLE FROM THE SERVICES.

THE DOCUMENTS AND RELATED GRAPHICS CONTAINED HEREIN COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN. MICROSOFTAND/OR ITS RESPECTIVE SUPPLIERS MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED HEREIN AT ANY TIME. PARTIAL SCREEN SHOTS MAY BE VIEWED IN FULL WITHIN THE SOFTWARE VERSION SPECIFIED.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

MICROSOFT® WINDOWS®, AND MICROSOFT OFFICE® ARE REGISTERED TRADEMARKS OF THE MICROSOFT CORPORATION IN THE U.S.A. AND OTHER COUNTRIES. THIS BOOK IS NOT SPONSORED OR ENDORSED BY OR AFFILIATED WITH THE MICROSOFT CORPORATION.

About the Author

John Garrett is CCIE Emeritus (6204) and Splunk Certified. He earned an M.S. in predictive analytics from Northwestern University, and has a patent pending related to analysis of network devices with data science techniques. John has architected, designed, and implemented LAN, WAN, wireless, and data center solutions for some of the largest Cisco customers. As a secondary role, John has worked with teams in the Cisco Services organization to innovate on some of the most widely used tools and methodologies at Customer Experience over the past 12 years.

For the past 7 years, John's journey has moved through server virtualization, network virtualization, OpenStack and cloud, network functions virtualization (NFV), service assurance, and data science. The realization that analytics and data science play roles in all these brought John full circle back to developing innovative tools and techniques for Cisco Services. John's most recent role is as an Analytics Technical Lead, developing use cases to benefit Cisco Services customers as part of Business Critical Services for Cisco. John lives with his wife and children in Raleigh, North Carolina.

About the Technical Reviewers

Dr. Ammar Rayes is a Distinguished Engineer at Advance Services Technology Office for Cisco, focusing on network analytics, IoT, and machine learning. He has authored 3 books and more than 100 publications in refereed journals and conferences on advances in software- and networking-related technologies, and he holds more than 25 patents. He is the founding president and board member of the International Society of Service Innovation Professionals (www.issip.org), editor-in-chief of the journal *Advancements in Internet of Things* and an editorial board member of the European Alliance for Innovation—Industrial Networks and Intelligent Systems. He has served as associate editor on the journals *ACM Transactions on Internet Technology* and *Wireless Communications and Mobile Computing* and as guest editor on multiple journals and several *IEEE Communications Magazine* issues. He has co-chaired the Frontiers in Service conference and appeared as keynote speaker at several IEEE and industry conferences.

At Cisco, Ammar is the founding chair of Cisco Services Research and the Cisco Services Patent Council. He received the Cisco Chairman's Choice Award for IoT Excellent Innovation and Execution.

He received B.S. and M.S. degrees in electrical engineering from the University of Illinois at Urbana and a Ph.D. in electrical engineering from Washington University in St. Louis, Missouri, where he received the Outstanding Graduate Student Award in Telecommunications.

Nidhi Kao is a Data Scientist at Cisco Systems who develops advanced analytic solutions for Cisco Advanced Services. She received a B.S. in biochemistry from North Carolina State University and an M.B.A. from the University of North Carolina Kenan Flagler Business School. Prior to working at Cisco Systems, she held analytic chemist and research positions in industry and nonprofit laboratories.

Dedications

This book is dedicated to my wife, Veronica, and my children, Lexy, Trevor, and Mason. Thank you for making it possible for me to follow my passions through your unending support.

Acknowledgments

I would like to thank my manager, Ulf Vinneras, for supporting my efforts toward writing this book and creating an innovative culture where Cisco Services incubation teams can thrive and grow.

To that end, thanks go out to all the people in these incubation teams in Cisco Services for their constant sharing of ideas and perspectives. Your insightful questions, challenges, and solutions have led me to work in interesting roles that make me look forward to coming to work every day. This includes the people who are tasked with incubation, as well as the people from the field who do it because they want to make Cisco better for both employees and customers.

Thank you, Nidhi Kao and Ammar Rayes, for your technical expertise and your time spent reviewing this book. I value your expertise and appreciate your time. Your recommendations and guidance were spot-on for improving the book.

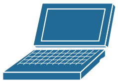
Finally, thanks to the Pearson team for helping me make this career goal a reality. There are many areas of publishing that were new to me, and you made the process and the experience very easy and enjoyable.

Contents at a Glance

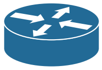
Contents

Icons Used in This Book

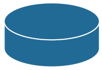
Cloud-line



Laptop



Router



Relational Database

Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Foreword

What's the future of network engineers? This is a question haunting many of us. In the past, it was somewhat easy; study for your networking certification, have the CCIE or CCDE as the ultimate goal, and your future was secured.

In my job as a General Manager within the Cisco Professional Services organization, working with Fortune 1000 clients from around the world, I meet a lot of people with opinions in this matter, with views ranging from "we just need software programmers in the future" to "data scientist is the way to go as we will automate everything." Is either of these views correct?

My simple answer to this is, "no," the long answer is a little more complicated.

The changes in the networking industry are to a large extent the same as the automotive industry; today most cars are computerized. Imagine though, if a car was built by people that only knew software programming, and didn't know anything about the car design, the engine, or security. The "architect" of a car needs to be an in-depth expert on car design, and at the same time know enough about software capabilities, and what can be achieved, in a way that still keeps the "soul" of the car and enhances the overall result.

When it comes to the future of networking, it is very much the same. If we replaced skilled network engineers with data science engineers, the result would be mediocre. At the same time, there is no doubt that the future of networking will be built on data science.

In my view, the ideal structure of any IT team is a core of very knowledgeable network engineers, working very closely together with skilled data scientists. The network engineers that take the time to learn the basics of data science, and start to expand into that area will automatically be the bridge to the data science, and these engineers will soon become the most critical asset in that IT department.

The author of this book, John Garrett, is a true example of someone that has made this journey. With many years of experience working with the largest Cisco clients around the world, as one of our more senior network and data center technical leads, John saw the movement of data science approaching, and decided to invest himself in learning this new discipline. I would say he did not only learn it but instead mastered the art.

In this book, John helps the reader along the journey of learning data analytics in a very practical and applied way, providing the tools to almost immediately provide value to your organization.

At the end of the day, career progress is very linked to providing unique value. If you have decided to invest in yourself, and build data science skills on top of your telecommunication, datacenter, security, or IT knowledge, this book is the perfect start.

I would argue that John is a proof point to this matter, moving from a tech lead consultant to now being part of a small core team focusing on innovation to create the future of professional services from Cisco. A confirmation of this is also the number of patent submissions that John has pending in the area, as networking skills combined with data science opened up entirely new avenues of capabilities and solutions.

By Ulf Vinneras, Cisco General Manager Customer Experience/Cross Architecture

Introduction: Your future is in your hands!

Analytics and data science are everywhere. Everything today is connected by networks. In the past networking and data science were distinct career paths, but this is no longer the case. Network and information technology (IT) specialists can benefit from understanding analytics, and data scientists can benefit from understanding how computer networks operate and produce data. People in both roles are responsible for building analytics solutions and use cases that improve the business.

This book provides the following:

- An introduction to data science methodologies and algorithms for network and IT professionals
- An understanding of computer network data that is available from these networks for data scientists
- Techniques for uncovering innovative use cases that combine the data science algorithms with network data
- Hands-on use-case development in Python and deep exploration of how to combine the networking data and data science techniques to find meaningful insights

After reading this book, data scientists will experience more success interacting with IT networking experts, and IT networking experts will be able to aid in developing complete analytics solutions. Experts from either area will learn how to develop networking use cases independently.

My Story

I am a network engineer by trade. Prior to learning anything about analytics, I was an engineer working in data networking. Thanks to my many years of experience, I could design most network architectures that used any electronics to move any kind of data—business critical or not—in support of world-class applications. I thought I knew everything I needed to know about networking.

Then digital transformation happened. The software revolution happened. Everything went software defined. Everything is “virtual” and “containerized” now. Analytics is everywhere. With all these changes, I found that I didn’t know as much as I once thought I did.

If this sounds like your story, then you have enough experience to realize that you need to understand the next big thing if you want to remain relevant in a networking-related role—and analytics applied in your networking domain of expertise is the next big thing for you. If yours is like many organizations today, you have tons of data, and you have analytics tools and software to dive into it, but you just do not really know what to do with it. How can your skills be relevant here? How do you make the connection from these buckets, pockets, and piles of data to solving problems for your company? How can you develop use cases that solve both business and technical problems? Which use cases provide some real value, and which ones are a waste of your time?

Looking for that next big thing was exactly the situation I found myself in about 10 years ago. I was experienced when it came to network design. I was a 5 year CCIE, and I had transitioned my skill set from campus design to wireless to the data center. I was working in one of the forward-looking areas of Cisco Services, Cisco Advanced Services. One of our many charters was "proactive customer support," with a goal of helping customers avoid costly outages and downtime by preventing problems from happening in the first place. While it was not called *analytics* back then, the work done by Cisco Advanced Services could fall into a bucket known today as *prescriptive analytics*.

If you are an engineer looking for that next step in your career, many of my experiences will resonate with you. Many years ago, I was a senior technical practitioner deciding what was next for developing my skill set. My son was taking Cisco networking classes in high school, and the writing was on the wall that being only a network engineer was not going to be a viable alternative in the long term. I needed to level up my skills in order to maintain a senior-level position in a networking-related field, or I was looking at a role change or a career change in the future.

Why analytics? I was learning through my many customer interactions that we needed do more with the data and expertise that we had in Cisco Services. The domain of coverage in networking was small enough back then that you could identify where things were "just not right" based on experience and intuition. At Cisco, we know how to use our collected data, our knowledge about data on existing systems, and our intuition to develop "mental models" that we regularly apply to our customer network environments.

What are mental models? Captain Sully on US Airways flight 1549 used mental models when he made an emergency landing on the Hudson River in 2009. Given all of the airplane telemetry data, Captain Sully knew best what he needed to do in order to land the plane safely and protect the lives of hundreds of passengers. Like experienced airplane pilots, experienced network engineers like you know how to avoid catastrophic failures. Mental models are powerful, and in this book, I tell you how to use mental models and innovation techniques to develop insightful analytics use cases for the networking domain.

The Services teams at Cisco had excellent collection and reporting. Expert analysis in the middle was our secret sauce. In many cases, the anonymized data from these systems became feeds to our internal tools that we developed as "digital implementations" of our mental models. We built awesome collection mechanisms, data repositories, proprietary rule-matching systems, machine reasoning systems, and automated reporting that we could use to summarize all the data in our findings for Cisco Services customers. We were finding insights but not actively looking for them using analytics and machine learning.

My primary interest as a futurist thinker was seeking to understand what was coming next for Cisco Advanced Services and myself. What was the "next big thing" for which we needed to be prepared? In this pursuit, I explored a wide array of new technology areas over the course of 10 years. I spent some years learning and designing VMware, OpenStack, network functions virtualization (NFV), and the associated virtual network functions (VNFs) solutions on top of OpenStack. I then pivoted to analytics and applied those concepts to my virtualization knowledge area.

After several years working on this cutting edge of virtualized software infrastructure design and analytics, I learned that whether the infrastructure is physical or virtual, whether the applications are local or in the cloud, the importance of being able to find insights within the data that we get from our networking environments is critical to the success of these environments. I also learned that the growth of data science and the availability of computer resources to munge through the data make analytics and data science very attainable for any networking professional who wishes to pivot in this direction.

Given this insight, I spent 3 years of time outside work, including many evenings, weekends, and all of my available vacation time in order to earn a master's degree in predictive analytics from Northwestern University. Around that same time I began reading (or listening to) hundreds of books, articles, and papers about analytics topics. I also consumed interesting writings about algorithms, data science, innovation, innovative techniques, brain chemistry, bias, and other topics related to turning data into value by using creative thinking techniques. You are an engineer, so you can associate this to learning that next new platform, software, or architecture. You go all in.

Another driver for me was that I am work centered, driven to succeed, and competitive by nature. Maybe you are, too. My customers who had purchased Cisco services were challenging us to do better. It was no longer good enough to say that everything is connected, traffic is moving just fine across your network, and if there is a problem, the network protocols will heal themselves. Our customers wanted more than that.

Cisco Advanced Services customers are highly skilled, and they wanted more than simple reporting. They wanted visibility and insights across many domains. My customers wanted data, and they wanted dashboards that shared data with them so they could determine what was wrong on their own. One customer (we will call him Dave because that was his name) wanted to be able to use his own algorithms, his own machines, and his own people to determine what was happening at the lower levels of his infrastructure. He wanted to correlate this network data with his applications and his business metrics. For me, as a very senior network and data center engineer, I felt like I was not getting the job done. I could not do the analytics. I did not have a solution that I could propose for his purpose. There was a new space in networking that I had not yet conquered. Dave wanted actionable intelligence derived from the data that he was providing to Cisco. Dave wanted real analytics insights. Challenge accepted.

That was the start of my journey into analytics and into making the transition from being a network engineer to being a data scientist with enough ability to bridge the gap between IT networking engineers and those mathematical wizards who do the hard-core data science. This book is a knowledge share of what I have learned over the past years as I have transitioned from being an enterprise-focused campus, WAN, and data center networking engineer to being a learning data scientist. I realized that it was not necessary to get to the Ph.D. level to use data science and predictive analytics. For my transition, I wanted to be someone who can use enough data science principles to find use cases in the wild and apply them to common IT networking problems to find useful, relevant, and actionable insights for my customers.

I hope you enjoy reading about what I have learned on this journey as much as I have enjoyed learning it. I am still working at it, so you will get the very latest. I hope that my learning and experiences in data, data science, innovation, and analytics use cases can help you in your career.

How This Book Is Organized

Chapter 1, "Getting Started with Analytics," defines some further details about what is explored in this book, as well as the current analytics landscape in the media. You cannot open your laptop or a social media application on your phone without seeing something related to analytics.

Chapter 2, "Approaches for Analytics and Data Science," explores methodologies and approaches that will help you find success as a data scientist in your area of expertise. The simple models and diagrams that I have developed for internal Cisco trainings can help with your own solution framing activities.

Chapter 3, "Understanding Networking Data Sources," begins by looking at network data and the planes of operation in networks that source this data. Virtualized solutions such as OpenStack and network functions virtualization (NFV) create additional complexities with sourcing data for analysis. Most network devices can perform multiple functions with the same hardware. This chapter will help you understand how they all fit together so you can get the right data for your solutions.

Chapter 4, "Accessing Data from Network Components," introduces networking data details. Networking environments produce many different types of data, and there are multiple ways to get at it. This chapter provides overviews of the most common data access methods in networking. You cannot be a data scientist without data! If you are a seasoned networking engineer, you may only need to skim this chapter.

Chapter 5, "Mental Models and Cognitive Bias," shifts gears toward innovation by spending time in the area of mental models, cognitive science, and bias. I am not a psychology expert or an authority in this space, but in this chapter I share common biases that you may experience in yourself, your users, and your stakeholders. This cognitive science is where things diverge from a standard networking book—but in a fascinating way. Understanding your audience is key to building successful use cases for them.

Chapter 6, "Innovative Thinking Techniques," introduces innovative techniques and interesting tricks that I have used to uncover use cases in my role with Cisco. Understanding bias from Chapter 5 coupled with innovation techniques from this chapter will prepare you to maximize the benefit of the use cases and algorithms you learn in the upcoming chapters.

Chapter 7, "Analytics Use Cases and the Intuition Behind Them," has you use your new knowledge of innovation to walk through analytics use cases across many industries. I have learned that combining the understanding of data with new and creative—and sometimes biased—thinking results in new understanding and new perspective.

Chapter 8, "Analytics Algorithms and the Intuition Behind Them," walks through many common industry algorithms from the use cases in Chapter 7 and examines the intuition behind them. Whereas Chapter 7 looks at use cases from a top-down perspective, this chapter looks at algorithms to give you an inside-out view. If you know the problems you want to solve, this is your toolbox.

Chapter 9, "Building Analytics Use Cases," brings back the models and methodologies from Chapter 2 and reviews how to turn your newfound ideas and algorithms into solutions. The use cases and data for the next four chapters are outlined here.

Chapter 10, "Developing Real Use Cases: The Power of Statistics," moves from the abstract to the concrete and explores some real Cisco Services use cases built around statistics. There is still a very powerful role for statistics in our fancy data science world.

Chapter 11, "Developing Real Use Cases: Network Infrastructure Analytics," looks at actual solutions that have been built using the feature information about your network infrastructure. A detailed look at Cisco Advanced Services fingerprinting, and other infrastructure-related capabilities is available here.

Chapter 12, "Developing Real Use Cases: Control Plane Analytics Using Syslog Telemetry," shows how to build solutions that use network event telemetry data. The popularity of pushing data from devices is growing, and you can build use cases by using such data. Familiar algorithms from previous chapters are combined with new data in this chapter to provide new insight.

Chapter 13, "Developing Real Use Cases: Data Plane Analytics," introduces solutions built for making sense of data plane traffic. This involves analysis of the packets flowing across your network devices. Familiar algorithms are used again to show how you can use the same analytics algorithms in many ways on many different types of data to find different insights.

Chapter 14, "Cisco Analytics," runs through major Cisco product highlights in the analytics space. Any of these products can function as data collectors, sources, or engines, and they can provide you with additional analytics and visualization capabilities to use for solutions that extend the capabilities and base offerings of these platforms. Think of them as "starter kits" that help you get a working product in place that you can build on in the future.

Chapter 15, "Book Summary," closes the book by providing a complete wrap-up of what I hope you learned as you read this book.

Credits

Stephen R. Covey, *The 7 Habits of Highly Effective People: Powerful Lessons in Personal Change*, 2004, Simon and Schuster.

ITU Annual Regional Human Capacity Building Workshop for Sub-Saharan Countries in Africa Mauritius, 28–30 June 2017

Empirical Model-Building and Response Surfaces, 1987, George box, John Wiley.

Predictably Irrational: The Hidden Forces that Shape Our Decisions, Dan Ariely, HarperCollins.

Thinking, Fast and Slow, Daniel Kahneman, Macmillan Publishers

Abraham Wald

Thinking, Fast and Slow, Daniel Kahneman, Macmillan Publishers

Thinking, Fast and Slow, Daniel Kahneman, Macmillan Publishers

Thinking, Fast and Slow, Daniel Kahneman, Macmillan Publishers

Charles Duhigg

De, B. E. (1985). Six thinking hats. Boston: Little, Browne and Company.

Henry Ford

Ries, E. (2011). *The lean startup: How constant innovation to creates radically successful businesses*. Penguin Books

The Post-Algorithmic Era Has Arrived By Bill Franks, Dec 14, 2017.

Figure Credits

Figure 8-13	Scikit-learn
Figure 8-32	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 8-33	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 8-34	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-07	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-08	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-18	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-22	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-23	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-24	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-26	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-27	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-30	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-31	Screenshot of Jupyter Notebook © 2018 Project Jupyter
Figure 10-32	Screenshot of Jupyter Notebook © 2018 Project Jupyter

Chapter 2

Approaches for Analytics and Data Science

This chapter examines a simple methodology and approach for developing analytics solutions. When I first started analyzing networking data, I used many spreadsheets, and I had a lot of data access, but I did not have a good methodology to approach the problems. You can only sort, filter, pivot, and script so much when working with a single data set in a spreadsheet. You can spend hours, days, or weeks diving into the data, slicing and dicing, pivoting this way and that…only to find that the best you can do is show the biggest and the smallest data points. You end up with no real insights. When you share your findings to glassy-eyed managers, the rows and columns of data are a lot more interesting to you than they are to them. I have learned through experience that you need more.

Analytics solutions look at data to uncover stories about what is happening now or what will be happening in the future. In order to be effective in a data science role, you must step up your storytelling game. You can show the same results in different ways—sometimes many different ways—and to be successful, you must get the audience to see what you are seeing. As you will learn in Chapter 5, “Mental Models and Cognitive Bias,” people have biases that impact how they receive your results, and you need to find a way to make your results relevant to each of them—or at least make your results relevant to the stakeholders who matter.

You have two tasks here. First, you need to find a way to make your findings interesting to nontechnical people. You can make data more interesting to nontechnical people with statistics, top-*n* reporting, visualization, and a good storyline. I always call this the “BI/BA of analytics,” or the simple descriptive analytics. Business intelligence (BI)/ business analytics (BA) dashboards are a useful form of data presentation, but they typically rely on the viewer to find insight. This has value and is useful to some extent but generally tops out at cool visualizations that I call “*Sesame Street* analytics.”

If you are from my era, you grew up with the *Sesame Street* PBS show, which had a segment that taught children to recognize differences in images and had the musical tagline “One of these things is not like the others.” Visualizations with anomalies identified in contrasting colors immediately help the audience see how "one of these things is not like the others," and you do not need a story if you have shown this properly. People look at your visualization or infographic and just see it.

Your second task is to make the data interesting to the technical people, your new data science friends, your peers. You do this with models and analytics, and your visualizing and storytelling must be at a completely new level. If you present "*Sesame Street* analytics" to a technical audience, you are likely to hear "That's just visualization; I want to know *why* is it an outlier." You need to do more—with real algorithms and analytics—to impress this audience. This chapter starts your journey toward impressing both audiences.

Model Building and Model Deployment

As mentioned in Chapter 1, "Getting Started with Analytics," when it comes to analytics models, people often overlook a very important distinction between *developing and building* and *implementing and deploying* models. The ability for your model to be usable outside your own computer is a critical success factor, and you need to know how to both build and deploy your analytics use cases. It is often the case that you build models centrally then deploy them at the edge of a network or at many edges of corporate or service provider networks. Where do you think the speech recognition models on your mobile phone were built? Where are they ultimately deployed? If your model is going to have impact in your organization, you need to develop workflows that use your model to benefit the business in some tangible way.

Many models are developed or built from batches of test data, perhaps with data from a lab or a big data cluster, built on users' machines or inside an analytics package of data science algorithms. This data is readily available, cleaned, and standardized, and they have no missing values. Experienced data science people can easily run through a bunch of algorithms to visualize and analyze the data in different ways to glean new and interesting findings. With this captive data, you can sometimes run through hundreds of algorithms with different parameters, treating your model like a black box, and only viewing the results. Sometimes you get very cool-looking results that are relevant. In the eyes of management or people who do not understand the challenges in data science, such development activity looks like the simple layout in Figure 2-1, where data is simply combined with data science to develop a solution. Say hello to your nontechnical audience. This is not a disparaging remark; some people—maybe even most people—prefer to just get to the point, and nothing gets to the point better than results. These people do not care about the details that you needed to learn in order to provide solutions at this level of simplicity.

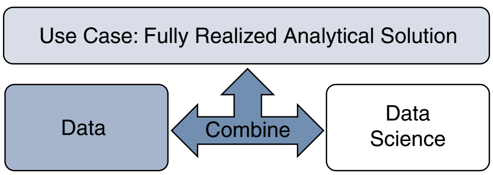


Figure 2-1 *Simplified View of Data Science*

Once you find a model, you bring in more data to further test and validate that the model's findings are useful. You need to prove beyond any reasonable doubt that the model you have on your laptop shows value. Fantastic. Then what? How can you bring all data across your company to your computer so that you can run it through the model you built?

At some point in the process, you will deploy your analytics to a production system, with real data, meaning that an automated system is set up to run new data, in batches or streaming, against your new model. This often involves working with a development team, whose members may or may not be experts in analytics. In some cases, you do not need to deploy into production at all because the insight is learned, and no further understanding is required. In either case, you then need to use your model against new batches of data to extend the value beyond the data you originally used to build and test it.

Because I am often the one with models on my computer, and I have learned how to deploy those models as part of useful applications, I share my experiences in turning models into useful tools in later chapters of this book, as we go through actual use cases.

Analytics Methodology and Approach

How you approach an analytics problem is one of the factors that determine how successful your solution will be in solving the problem. In the case of analytics problems, you can use two broad approaches, or methodologies, to get to insightful solutions. Depending on your background, you will have some predetermined bias in terms of how you want to approach problems. The ultimate goal is to convert data to value for your company. You get to that value by finding insights that solve technical or business problems. The two broad approaches, shown in Figure 2-2, are the "explore the data" approach, and the "solve the business problem" approach.

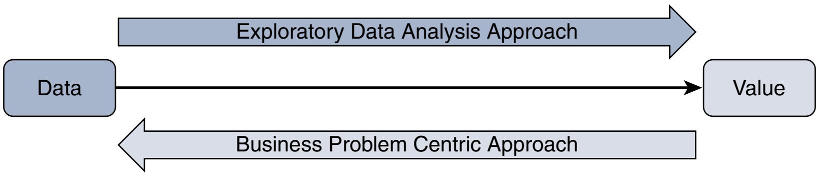


Figure 2-2 *Two Approaches to Developing Analytics Solutions*

These are the two main approaches that I use, and there is literature about many granular, systematic methodologies that support some variation of each of these approaches. Most analytics literature guides you to the problem-centric approach. If you are strongly aware of the data that you have but not sure how to use it to solve problems, you may find yourself starting in the statistically centered exploratory data analysis (EDA) space that is most closely associated with statistician John Tukey. This approach often has some quick wins along the way in finding statistical value in the data rollups and visualizations used to explore the data.

Most domain data experts tend to start with EDA because it helps you understand the data and get the quick wins that allow you to throw a bone to the stakeholders while digging into the more time-consuming part of the analysis. Your stakeholders often have hypotheses (and some biases) related to the data. Early findings from this side often sound like "You can see that issue *X* is highly correlated with condition *Y* in the environment; therefore, you should address condition *Y* to reduce the number of times you see issue *X*." Most of my early successes in developing tools and applications for Cisco Advanced Services were absolutely data first and based on statistical findings instead of analytics models. There were no heavy algorithms involved, there was no machine learning, and there was no real data science. Sometimes, statistics are just as effective at telling interesting stories. Figure 2-3 shows how to view these processes as a comparison. There is no right or wrong side on which to start; depending on your analysis goals, either direction or approach is valid. Note that this model includes data acquisition, data transport, data storage, sharing, or streaming, and secure access to that data, all of which are things to consider if the model is to be implemented on a production data flow—or "operationalized." The previous, simpler model that shows a simple data and data science combination (refer to Figure 2-1) still applies for exploring a static data set or stream that you can play back and analyze using offline tools.

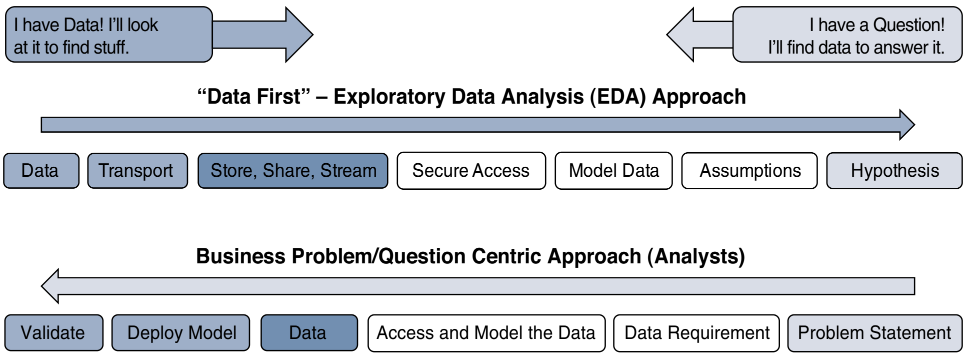


Figure 2-3 *Exploratory Data Versus Problem Approach Comparison*

Common Approach Walkthrough

While many believe that analytics is done only by math PhDs and statisticians, general analysts and industry subject matter experts (SMEs) now commonly use software to explore, predict, and preempt business and technical problems in their areas of expertise. You and other "citizen data scientists" can use a variety of software packages available today to find interesting insights and build useful models. You can start from either side when you understand the validity of both approaches. The important thing to understand is that many of the people you work with may be starting at the other end of the spectrum, and you need to be aware of this as you start sharing your insights with a wider audience. When either audience asks, "What problem does this solve for us?" you can present relevant findings.

Let's begin on the data side. During model *building*, you skip over the transport, store, and secure phases as you grab a batch of useful data, based on your assumptions, and try to test some hypothesis about it. Perhaps through some grouping and clustering of your trouble ticket data, you have seen excessive issues on your network routers with some specific version of software. In this case, you can create an analysis that proves your hypothesis that the problems are indeed related to the version of software that is running on the suspect network routers. For the data first approach, you need to determine the problems you want to solve, and you are also using the data to guide you to what is possible, given your knowledge of the environment.

What do you need in this suspect routers example? Obviously, you must get data about the network routers when they showed the issue, as well as data about the same types of routers that have not had the issue. You need both of these types of information in order to find the underlying factors that may or may not have contributed to the issue you are researching. Finding these factors is a form of inference, as you would like to infer something about all of your routers, based on comparisons of differences in a set of devices that exhibit the issue and a set of devices that do not. You will later use the same analytics model for prediction.

You can commonly skip the "production data" acquisition and transport parts of the model building phase. Although in this case you have a data set to work with for your analysis, consider here how to automate the acquisition of data, how to transport it, and where it will live if you plan to put your model into a fully automated production state so it can notify you of devices in the network that meet these criteria. On the other hand, full production state is not always necessary. Sometimes you can just grab a batch of data and run it against something on your own machine to find insights; this is valid and common. Sometimes you can collect enough data about a problem to solve that problem, and you can gain insight without having to implement a full production system.

Starting at the other end of this spectrum, a common analyst approach is to start with a known problem and figure out what data is required to solve that problem. You often need to seek things that you don't know to look for. Consider this example: Perhaps you have customers with service-level agreements (SLAs), and you find that you are giving them discounts because they are having voice issues over the network and you are not meeting the SLAs. This is costing your company money. You research what you need to analyze in order to understand why this happens, perhaps using voice drop and latency data from your environment. When you finally get these data, you build a proposed model that identifies that higher latency with specific versions of software on network routers is common on devices in the network path for customers who are asking for refunds. Then you deploy the model to flag these "SLA suckers" in your production systems and then validate that the model is effective as the SLA issues have gone away. In this case, *deploy* means that your model is watching your daily inventory data and looking for a device that matches the parameters that you have seen are problematic. What may have been a very complex model has a simple deployment.

Whether starting at data or at a business problem, ultimately solving the problem represents the value to your company and to you as an analyst. Both of these approaches follow many of the same steps on the analytics journey, but they often use different terminology. They are both about turning data into value, regardless of starting point, direction, or approach. Figure 2-4 provides a more detailed perspective that illustrates that these two approaches can work in the same environment on the same data and the very same problem statement. Simply put, all of the work and due diligence needs to be done to have a fully operational (with models built, tested, and deployed), end-to-end use case that provides real, continuous value.

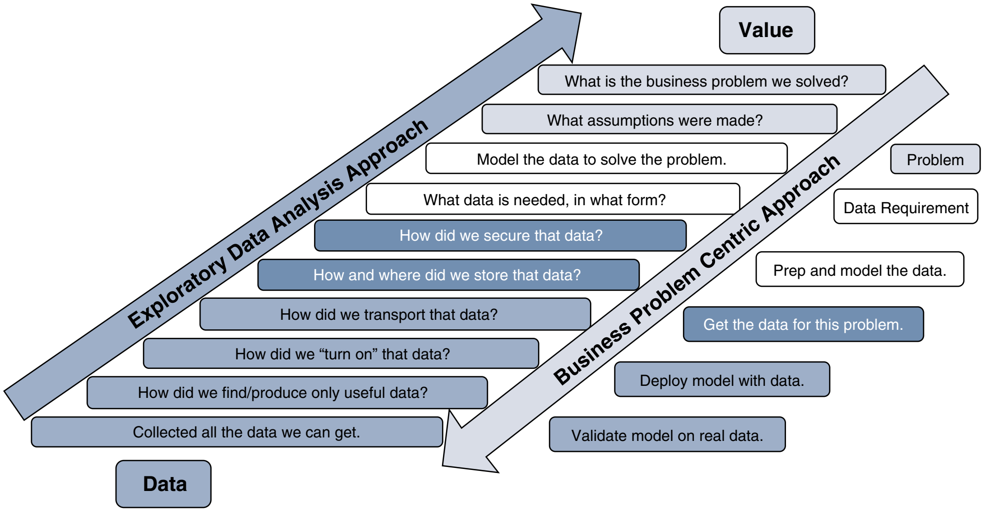


Figure 2-4 *Detailed Comparison of Data Versus Problem Approaches*

There are a wide variety of detailed approaches and frameworks available in industry today, such as CRISP-DM (cross-industry standard process for data mining) and SEMMA (Sample Explore, Modify, Model, and Assess), and they all generally follow these same principles. Pick something that fits your style and roll with it. Regardless of your approach, the primary goal is to create useful solutions in your problem space by combining the data you have with data science techniques to develop use cases that bring insights to the forefront.

Distinction Between the Use Case and the Solution

Let's slow down a bit and clarify a few terms. Basically, a *use case* is simply a description of a problem that you solve by combining data and data science and applying analytics. The underlying algorithms and models comprise the actual analytics solution. In the case of Amazon, for example, the use case is getting you to spend more money. Amazon does this by showing you what other people have also bought in addition to buying the same item that are purchasing. The intuition behind this is that you will buy more things because other people like you needed those things when they purchased the same item that you did. The model is there to uncover that and remind you that you may also need to purchase those other things. Very helpful, right?

From the exploratory data approach, Amazon might want to do something with the data it has about what people are buying online. It can then collect the high patterns of common sets of purchases. Then, for patterns that are close but missing just a few items, Amazon may assume that those people just “forgot” to purchase something they needed because everyone else purchased the entire “item set” found in the data. Amazon might then use software implementation to find the people who “forgot” and remind them that they might need the other common items. Then Amazon can validate the effectiveness by tracking purchases of items that the model suggested.

From a business problem approach, Amazon might look at wanting to increase sales, and it might assume (or find research which suggests) that, if reminded, people often purchase common companion items to what they are currently viewing or have in their shopping carts. In order to implement this, Amazon might collect buying pattern data to determine these companion items. The company might then suggest that people may also want to purchase these items. Amazon can then validate the effectiveness by tracking purchases of suggested items.

Do you see how both of these approaches reach the same final solution?

The Amazon case is about increasing sales of items. In predictive analytics, the use case may be about predicting home values or car values. More simply, the use case may be the ability to predict a continuous number from historical numbers. No matter the use case, you can view analytics as simply the application of data and data science to the problem domain. You can choose how you approach finding and building the solutions either by using the data as a guide or by dissecting the stated problem.

Logical Models for Data Science and Data

This section discusses analytics solutions that you model and build for the purpose of deployment to your environment. When I was working with Cisco customers in the early days of analytics, it became clear that setting up the entire data and data science pipeline as a working application on a production network was a bit confusing to many customers, as well as to traditional Cisco engineers.

Many customers thought that they could simply buy network analytics software and install it onto the network as they would any other application—and they would have fully insightful analytics. This, of course, is not the case. Analytics packages integrate into the very same networks for which you build models to run. We can use this situation to introduce the concept of an *overlay*, which is a very important concept for understanding network data (covered in Chapter 3, “Understanding Networking Data Sources”). Analytics packages installed on computers that sit on networks can *build* the models as discussed earlier, but when it is time to *deploy* the models that include data feeds from network environments, the analytics packages often have tendrils that reach deep into the network and IT systems. Further, these solutions can interface with business and customer data systems that exist elsewhere in the network. Designing such a system can be daunting because most applications on a network do not interact with the underlying hardware. A second important term you should understand is the *underlay*.

Analytics as an Overlay

So how do data and analytics applications fit within network architectures? In this context, you need to know the systems and software that consume the data, and you need to use data science to provide solutions as general applications. If you are using some data science packages or platforms today, then this idea should be familiar to you. These applications take data from the infrastructure (perhaps through a central data store) and combine it with other applications data from systems that reside within the IT infrastructure.

This means the solution is analyzing the very same infrastructure in which it resides, along with a whole host of other applications. In networking, an *overlay* is a solution that is abstracted from the underlying physical infrastructure in some way. Networking purists may not use the term *overlay* for applications, but it is used here because it is an important distinction needed to set up the data discussion in the next chapter. Your model, when implemented in production on a live network, is just an overlay instance of an application, much like other overlay application instances riding on the same network.

This concept of network layers and overlay/underlay is why networking is often blamed for fault or outage—because the network underlays all applications (and other network instances, as discussed in the next chapter). Most applications, if looked at from an application-centric view, are simply overlays onto the underlying network infrastructure. New networking solutions such as Cisco Application Centric Infrastructure (ACI) and common software-defined wide area networks (SD-WANs) such as Cisco iWAN+Viptela take overlay networking to a completely new level by adding additional layers of policy and network segmentation. In case you have not yet surmised, you probably should have a rock-solid underlay network if you want to run all these overlay applications, virtual private networks (VPNs), and analytics solutions on it.

Let's look at an example here to explain overlays. Consider your very own driving patterns (or walking patterns, if you are urban) and the roads or infrastructure that you use to get around. You are one overlay on the world around you. Your neighbor traveling is another overlay. Perhaps your overlay is "going to work," and your neighbor's overlay for the day is "going shopping." You are both using the same infrastructure but doing your own things, based on your interactions with the underlay (walkways, roads, bridges, home, offices, stores, and anything else that you interact with). Each of us is an individual "instance" using the underlay, much as applications are instances on networks. There could be hundreds or even thousands of these applications—or millions of people using the roadway system. The underlay itself has lots of possible "layers," such as the physical roads and intersections and the controls such as signs and lights. Unseen to you, and therefore "virtual," is probably some satellite layer where GPS is making decisions about how another application overlay (a delivery truck) should be using the underlay (roads).

This concept of overlays and layers, both physical and virtual, for applications as well as networks, was a big epiphany for me when I finally got it. The very networks themselves have layers and planes of operations. I recall it just clicking one day that the packets (routing protocol packets) that were being used to "set up" packet forwarding for a path in my network were using the same infrastructure that they were actually setting up. That is like me controlling the stoplights and walk signs as I go to work, while I am trying to get there. We'll talk more about this "control plane" later. For now, let's focus on what is involved with an analytics infrastructure overlay model.

By now, I hope that I have convinced you that this concept of some virtual overlay of functionality on a physical set of gear is very common in networking today. Let's now look at an analytics infrastructure overlay diagram to illustrate that the data and data science come together to form the use cases of always-on models running in your IT environment. Note in Figure 2-5 how other data, such as customer, business, or operations data, is exported from other application overlays and imported into yours.

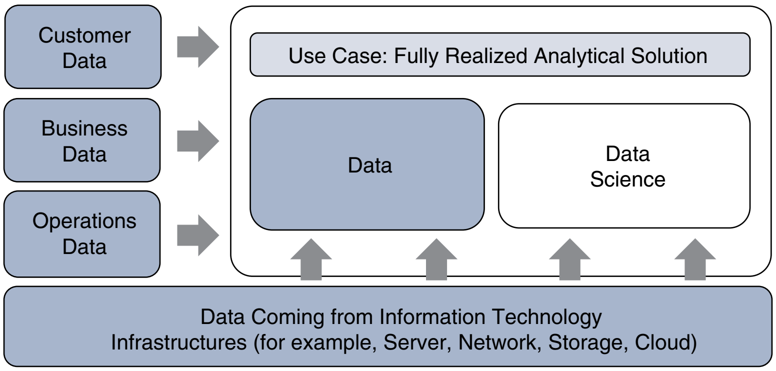


Figure 2-5 *Analytics Solution Overlay*

In today's digital environment, consider that all the data you need for analysis is produced by some system that is reachable through a network. Since everyone is connected, this is the very same network where you will use some system to collect and store this data. You will most likely deploy your favorite data science tools on this network as well. Your role as the analytics expert here is to make sure you identify how this is set up, such that you successfully set up the data sources that you need to build your analytics use case. You must ensure these data sources are available to the proper layer—your layer—of the network.

The concept of customer, business, and operations data may be new, so let's get right to the key value. If you used analytics in your customer space, you know who your valuable customers are (and, conversely, which customers are more costly than they are worth). This adds context to findings from the network, as does the business context (which network components have the greatest impact) and operations (where you are spending excessive time and money in the network). Bringing all these data together allows you to develop use cases with relevant context that will be noticed by business sponsors and stakeholders at higher levels in your company.

As mentioned earlier in this chapter, you can build a model with batches of data, but deploying an active model into your environment requires planning and setup of the data sources needed to "feed" your model as it runs every day in the environment. This may also include context data from other customer or business applications in the network environment. Once you have built a model and wish to operationalize it, making sure that everything properly feeds into your data pipelines is crucial—including the customer, business, operations, and other applications data.

Analytics Infrastructure Model

This section moves away from the overlays and network data to focus entirely on building an analytics solution. (We revisit the concepts of layers and overlays in the next chapter, when we dive deeper into the data sources in the networking domain.) In the case of IT networking, there are many types of deep technical data sources coming up from the environment, and you may need to combine them with data coming from business or operations systems in a common environment in order to provide relevance to the business. You use this data in the data science space with maturity levels of usage, as discussed in Chapter 1. So how can you think about data that is just "out there in the ether" in such a way that you can get to actual analytics use cases? All this is data that you define or create. This is just one component of a model that looks at the required data and components of the analytics use cases.

Figure 2-6 is a simple model for thinking about the flow of data for building deployable, operationalized models that provide analytics solutions. We can call this a simple model for analytics infrastructure, and, as shown in the figure, we can contrast this model with a problem-centric approach used by a traditional business analyst.

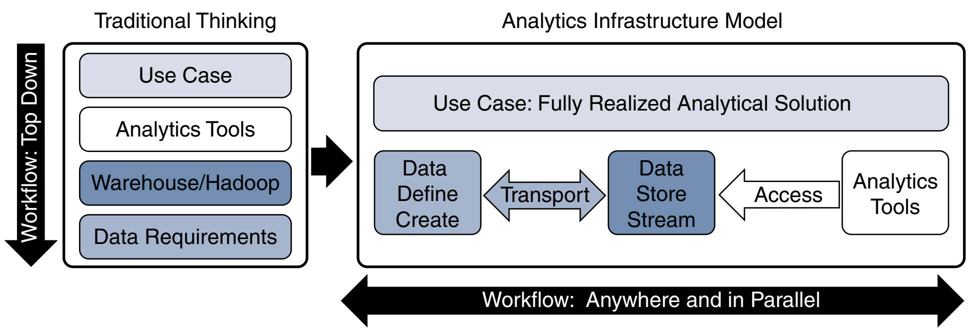


Figure 2-6 *Traditional Analyst Thinking Versus Analytics Infrastructure Model*

No, analytics infrastructure is not artificial intelligence. Due to the focus on the lower levels of infrastructure data for analytics usage, this analytics infrastructure name fits best. The goal is to identify how to build analytics solutions much the same way you have built LAN, WAN, wireless, and data center network infrastructures for years. Assembling a full architecture to extract value from data to solve a business problem is an infrastructure in itself. This is very much like an end-to-end application design or an end-to-end networking design, but with a focus on analytics solutions only.

The analytics infrastructure model used in IT networking differs from traditional analyst thinking in that it involves always looking to build repeatable, reusable, flexible solutions and not just find a data requirement for a single problem. This means that once you set up a data source—perhaps from routers, switches, databases, third-party systems, network collectors, or network management systems—you want to use that data source for multiple applications. You may want to replicate that data pipeline across other components and devices so others in the company can use it. This is the "build once, use many" paradigm that is common in Cisco Services and in Cisco products. Solutions built on standard interfaces are connected together to form new solutions. These solutions are reused as many times as needed. Analytics infrastructure model components can be used as many times as needed.

It is important to use standards-based data acquisition technologies and perhaps secure the transport and access around the central data cleansing, sharing, and storage of any networking data. This further ensures the reusability of your work for other solutions. Many such standard data acquisition techniques for the network layer are discussed in Chapter 4, "Accessing Data from Network Components."

At the far right of the model in Figure 2-6, you want to use any data science tool or package you can to access and analyze your data to create new use cases. Perhaps one package builds a model that is implemented in code, and another package produces the data visualization to show what is happening. The components in the various parts of the model are pluggable so that parts (for example, a transport or a database) could be swapped out with suitable replacements. The role and functionality of a component, not the vendor or type, is what is important.

Finally, you want to be able to work this in an Agile manner and not depend on the top-down Waterfall methods used in traditional solution design. You can work in parallel in any sections of this analytics infrastructure model to help build out the components you need to enable in order to operationalize any analytics model onto any network infrastructure. When you have a team with different areas of expertise along the analytics infrastructure model components, the process is accelerated.

Later in the book, this model is referenced as an aid to solution building. The analytics infrastructure model is very much a generalized model, but it is open, flexible, and usable across many different job roles, both technical and nontechnical, and allows for discussion across silos of people with whom you need to interface. All components are equally important and should be used to aid in the design of analytics solutions.

The analytics infrastructure model (shown enlarged in in Figure 2-7) also differs from many traditional development models in that it segments functions by job roles, which allows for the aforementioned Agile parallel development work. Each of these job roles may still use specialized models within its own functions. For example, a data scientist might use a preferred methodology and analytics tools to explore the data that you provided in the data storage location. As a networking professional, defining and creating data (far left) in your domain of expertise is where you play, and it is equally as important as the setup of the big data infrastructure (center of the model) or the analysis of the data using specialized tools and algorithms (far right).

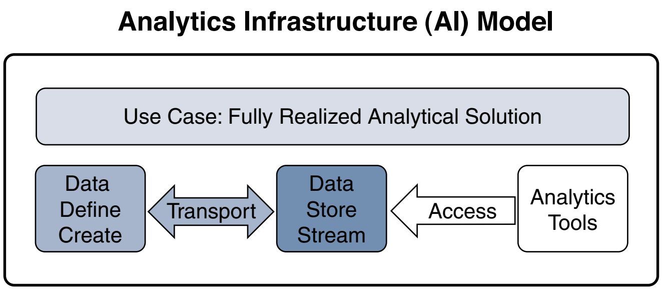


Figure 2-7 *Analytics Infrastructure Model for Developing Analytics Solutions*

Here is a simple elevator pitch for the analytics infrastructure model: "Data is defined, created, or produced in some system from which it is moved into a place where it is stored, shared, or streamed to interested users and data science consumers. Domain-specific solutions using data science tools, techniques, and methodologies provide the analysis and use cases from this data. A fully realized solution crosses all of the data, data storage, and data science components to deliver a use case that is relevant to the business."

As mentioned in Chapter 1, this book spends little time on "the engine," which is the center of this model, identified as the big data layer shown in Figure 2-8. When I refer to anything in this engine space, I call out the function, such as "store the data in a database" or "stream the data from the Kafka bus." Due to the number of open source and commercial components and options in this space, there is an almost infinite combination of options and instructions readily available to build the capabilities that you need.

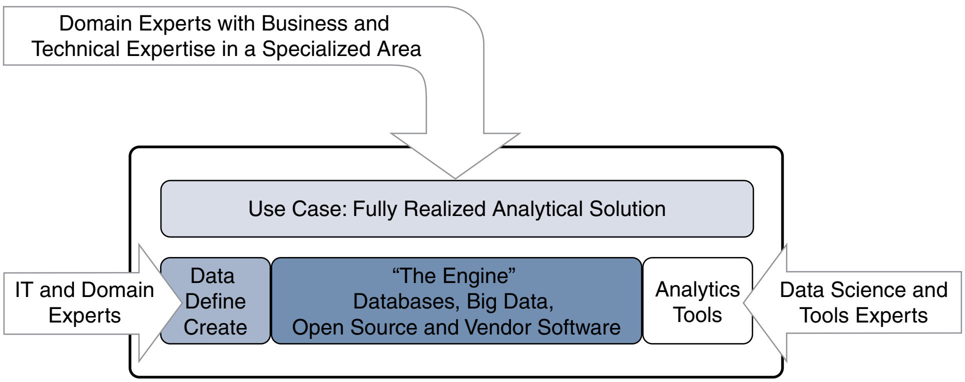


Figure 2-8 *Roles and the Analytics Infrastructure Model*

It is not important that you understand how “the engine” in this car works; rather, it is important to ensure that you can use it to drive toward analytics solutions. Whether using open source big data infrastructure or packages from vendors in this space, you can readily find instructions to transport, store, share, and stream and provide access to the data on the Internet. Run a web search on “data engineering pipelines” and “big data architecture,” and you will find a vast array of information and literature in the data engineering space.

The book aims to help you understand the job roles around the common big data infrastructure, along with data, data science, and use cases. The following are some of the key roles you need to understand:

- **Data domain experts—**These experts are familiar with the data and data sources.
- **Analytics or business domain experts—**These experts are familiar with the problems that need to be solved (or questions that need to be answered).
- **Data scientists—**These experts have knowledge of the tools and techniques available to find the answers or insights desired by the business or technical experts in the company.

The analytics infrastructure model is location agnostic, which is why you see callouts for data transport and data access. This overall model approach applies regardless of technology or location. Analytics systems can be on-premises, in the cloud, or hybrid solutions, as long as all the parts are available for use. Regardless of where the analytics is used, the networking team is a usually involved in ensuring that the data is in the right place for the analysis. Recall from the overlay discussion earlier in the chapter that the underlay is necessary for the overlay to work. Parts of this analysis may exist in the cloud, other parts on your laptop, and other parts on captive customer relationship management (CRM) systems on your corporate networks. You can use the analytics infrastructure model to diagram a solution flow that results in a fully realized analytics use case.

Depending on your primary role, you may be involved in gathering the data, moving the data, storing the data, sharing the data, streaming the data, archiving the data, or providing the analytics analysis. You may be ready to build the entire use case. There are many perspectives when discussing analytics solutions. Sometimes you will wear multiple hats. Sometimes you will work with many people; sometimes you will work alone if you have learned to fill all the required roles. If you decide to work alone, make sure you have access to resources or expertise to validate findings in areas that are new to you. You don’t want to spend a significant amount of time uncovering something that is already general knowledge and therefore not very useful to your stakeholders.

Building your components using the analytics infrastructure model ensures that you have reusable assets in each of the major parts of the model. Sometimes you will spend many hours, days, or weeks developing an analysis, only to find that there are no interesting insights. This is common in data science work. By using the analytics infrastructure model, you can maintain some parts of your work to build other solutions in the future.

The Analytics Infrastructure Model In Depth

So what are the "reusable and repeatable components" touted in the analytics infrastructure model? This section digs into the details of what needs to happen in each part of the model. Let's start by digging into the lower-left data component of the model, looking at the data that is commonly available in an IT environment. Data pipelines are big business and well covered in the "for fee" and free literature.

Building analytics models usually involves getting and modeling some data from the infrastructure, which includes spending a lot of time on research, data munging, data wrangling, data cleansing, ETL (Extract, Transform, Load), and other tasks. The true power of what you build is realized when you deploy your model into an environment and turn it on. As the analytics infrastructure model indicates, this involves acquiring useful data and transporting it into an accessible place. What are some examples of the data that you may need to acquire? Expanding on the data and transport sections of the model in Figure 2-9, you will find many familiar terms related to the combination of networking and data.

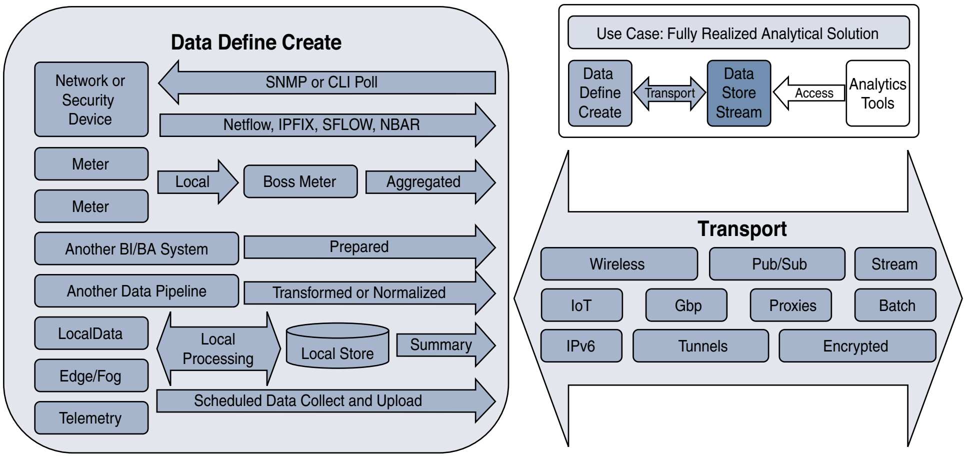


Figure 2-9 *Analytics Infrastructure Model Data and Transport Examples*

Implementing a model involves setting up a full pipeline of new data (or reusing a part of a previous pipeline) to run through your newly modeled use cases, and this involves "turning on" the right data and transporting it to where you need it to be. Sometimes this is kept local (as in the case of many Internet of Things [IoT] solutions), and sometimes data needs to be transported. This is all part of setting up the full data pipeline. If you need to examine data in flight for some real-time analysis, you may need to have full data streaming capabilities built from the data source to the place where the analysis happens.

Do not let the number of words in Figure 2-9 scare you; not all of these things are used. This diagram simply shares some possibilities and is in no way a complete set of everything that could be at each layer.

To illustrate how this model works, let's return to the earlier example of the router problem. If latency and sometimes router crashes are associated with a memory leak in some software versions of a network router, you can use a telemetry data source to access memory statistics in a router. Telemetry data, covered in Chapter 4, is a push model whereby network devices send periodic or triggered updates to a specified location in the analytics solution overlay. Telemetry is like a hospital heart monitor that gets constant updates from probes on a patient. Getting router memory–related telemetry data to the analytics layer involves using the components identified in white in Figure 2-10—for just a single stream. By setting this up for use, you create a reusable data pipeline with telemetry-supplied data. A new instance of this full pipeline must be set up for each device in the network that you want to analyze for this problem. The hard part—the "feature engineering" of building a pipeline—needs to happen only once. You can easily replicate and reuse that pipeline, as you now have your memory "heart rate monitor" set up for all devices that support telemetry. The left side of Figure 2-10 shows many ways data can originate, including methods and local data manipulations, and the arrow on the right side of the figure shows potential transport methods. There are many types of data sources and access methods.

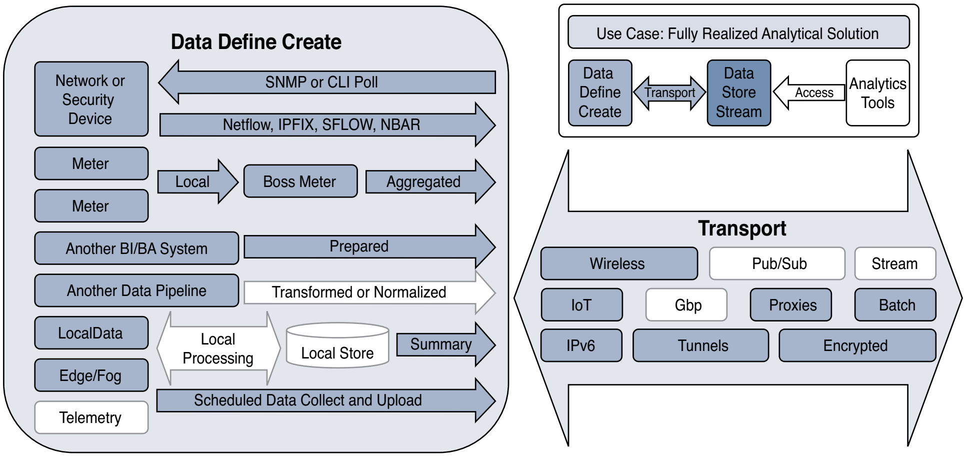


Figure 2-10 *Analytics Infrastructure Model Telemetry Data Example*

In this example, you are taking in telemetry data at the data layer, and you may also do some local processing of the data and store it in a localized database. In order to send the memory data upstream, you may standardize it to a megabyte or gigabyte number, standardize it to a "z" value, or perform some other transformation. This design work must happen once for each source. Does this data transformation and standardization stuff sound tedious to you? Consider that in 1999, NASA lost a $125 million Mars orbiter due to a mismatch of metric to English units in the software. Standardization, transformation, and data design are important.

Now, assuming that you have the telemetry data you want, how do you send it to a storage location? You need to choose transport options. For this example, say that you choose to send a steady stream to a Kafka publisher/subscriber location by using Google Protocol Buffers (GPB) encoding. There are lots of capabilities, and lots of options, but after a one-time design, learning, and setup process, you can document it and use it over and over again. What happens when you need to check another router for this same memory leak? You call up the specification that you designed here and retrofit it for the new requirement.

While data platforms and data movement are not covered in detail in this book, it is important that you have a basic understanding of what is happening inside the engine, all around the "the data platform."

The Analytics Engine

Unless you have a dedicated team to do this, much of this data storage work and setup may fall in your lap during model building. You can find a wealth of instruction for building your own data environments by doing a simple Internet search. Figure 2-11 shows many of the activities related to this layer. Note how the transport and data access relate to the configuration of this centralized engine. You need a destination for your prepared data, and you need to know the central location configuration so you can send it there. On the access side, the central data location will have access methods and security, which you must know or design in order to consume data from this layer.

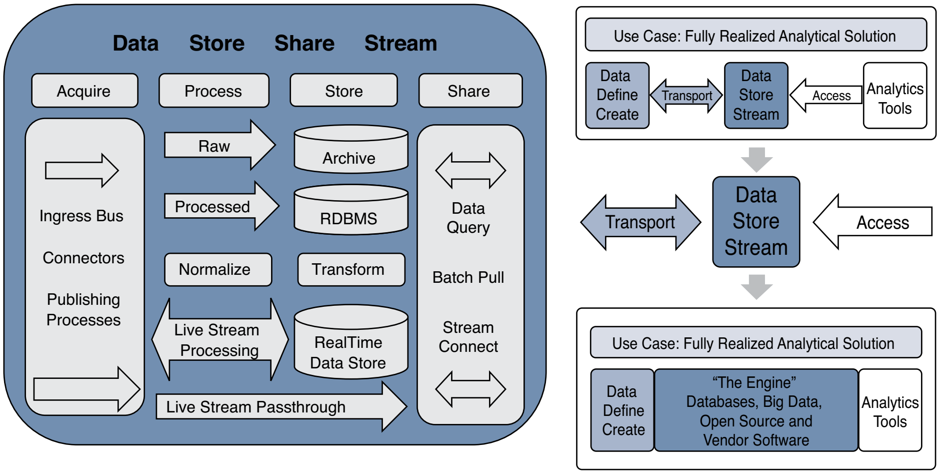


Figure 2-11 *The Analytics Infrastructure Model Data Engine*

Once you have defined the data parameters, and you understand where to send the data, you can move the data into the engine for storage, analysis, and streaming. From each individual source perspective, the choice comes down to push or pull mechanisms, as per the component capabilities available to you in your data-producing entities. This may include pull methods using polling protocols such as Simple Network Management Protocol (SNMP) or push methods such as the telemetry used in this example.

This centralized data-engineering environment is where the Hadoop, Spark, or commercial big data platform lives. Such platforms are often set up with receivers for each individual type of data. The pipeline definition for each of these types of data includes the type and configuration of this receiver at the central data environment. Very common within analytics engines today is something called a publisher/subscriber environment, or "pub/sub" bus. Apache Kafka is a very common bus used in these engines today.

A good analogy for the pub/sub bus is broadcast TV channels with a DVR. Data feeds (through analytics infrastructure model transports) are sent to specific channels from data producers, and subscribers (data consumers) can choose to listen to these data feeds and subscribe (using some analytics infrastructure model access method, such as a Kafka consumer) to receive them. In this telemetry example, the telemetry receiver takes interesting data and copies or publishes it to this bus environment. Any package requiring data for doing analytics subscribes to a stream and has it copied to its location for analysis in the case of streaming data. This separation of the data producers and consumers makes for very flexible application development. It also means that your single data feed could be simultaneously used by multiple consumers.

What else happens here at the central environment? There are receivers for just about any data type. You can both stream into the centralized data environment and out of the centralized environment in real time. While this is happening, processing functions decode the stream, extract interesting data, and put the data into relational databases or raw storage. It is also common to copy items from the data into some type of "object" storage environment for future processing. During the transform process, you may standardize, summarize, normalize, and store data. You transform data to something that is usable and standardized to fit into some existing analytics use case. This centralized environment, often called the "data warehouse" or "data lake," is accessed through a variety of methods, such as Structured Query Language (SQL), application programming interface (API) calls, Kafka consumers, or even simple file access, just to name a few.

Before the data is stored at the central location, you may need to adjust these data, including doing the following:

- Data cleansing to make sure the data matches known types that your storage expects
- Data reconciliation, including filling missing data, cleaning up formats, removing duplicates, or bounding values to known ranges
- Deriving or generating any new values that you want included in the records
- Splitting or combining data into meaningful values for the domain
- Standardizing the data ingress or splitting a stream to keep standardized and raw data

Now let's return to the memory example: These telemetry data streams (subject: memory leak) from the network infrastructure must now be made available to the analytics tools and data scientists for analysis or application of the models. This availability must happen through the analytics engine part of the analytics infrastructure model. Figure 2-12 shows what types of activities are involved if there is a query or request for this data stream from analytics tools or packages. This query is requesting that a live feed of the stream be passed through the publisher/subscriber bus architecture and a normalized feed of the same stream be copied to a database for batch analysis. This is all set up in the software at the central data location.

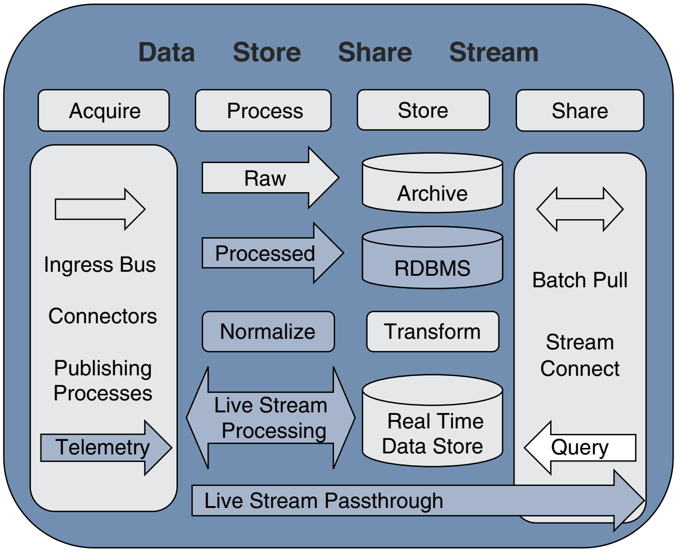


Figure 2-12 *Analytics Infrastructure Model Streaming Data Example*

Data Science

Data science is the sexy part of analytics. Data science includes the data mining, statistics, visualization, and modeling activities performed on readily available data. People often forget about the requirements to get the proper data to solve the individual use cases. The focus for most analysts is to start with the business problem first and then determine which type of data is required to solve or provide insights from the particular use case. Do not underestimate the time and effort required to set up the data for these use cases. Research shows that analysts spend 80% or more of their time on acquiring, cleaning, normalizing, transforming, or otherwise manipulating the data. I've spent upward of 90% on some problems.

Analysts must spend so much time because analytics algorithms require specific representations or encodings of the data. In some cases, encoding is required because the raw stream appears to be gibberish. You can commonly do the transformations, standardizations, and normalizations of data in the data pipeline, depending on the use case. First you need to figure out the required data manipulations through your model building phases; you will ultimately add them inline to the model deployment phases, as shown in the previous diagrams, such that your data arrives at the data science tools ready to use in the models.

The analytics infrastructure model is valuable from the data science tools perspective because you can assume that the data is ready, and you can focus clearly on the data access and the tools you need to work on that data. Now you do the data science part. As shown in Figure 2-13, the data science part of the model highlights tools, processes, and capabilities that are required to build and deploy models.

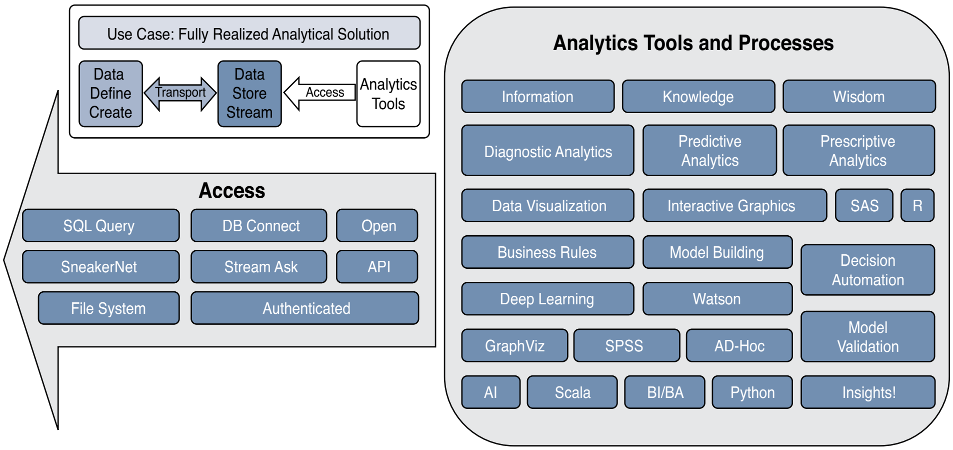


Figure 2-13 *Analytics Infrastructure Model Analytics Tools and Processes*

Going back to the streaming telemetry memory leak example, what should you do here? As highlighted in Figure 2-14, you use a SQL query to an API to set up the storage of the summary data. You also request full stream access to provide data visualization. Data visualization then easily shows both your technical and nontechnical stakeholders the obvious untamed growth of memory on certain platforms, which ultimately provides some "diagnostic analytics." Insight: This platform, as you have it deployed, leaks memory with the current network conditions. You clearly show this with a data visualization, and now that you have diagnosed it, you can even build a predictive model for catching it before it becomes a problem in your network.

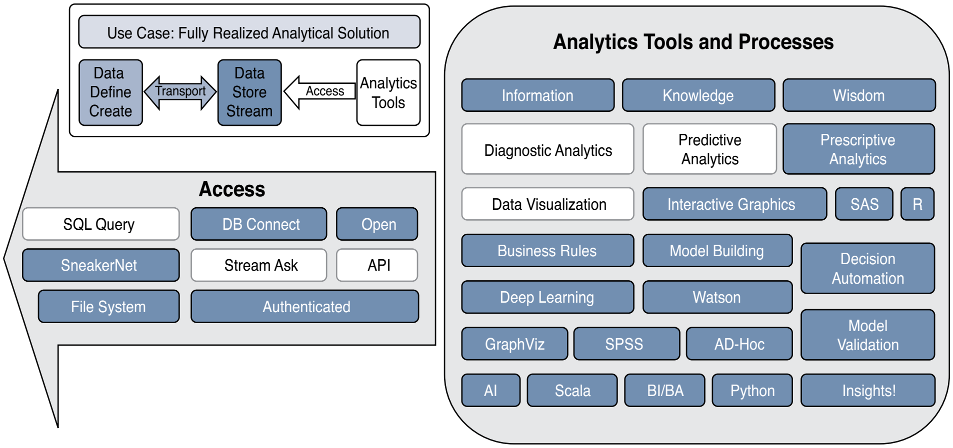


Figure 2-14 *Analytics Infrastructure Model Streaming Analytics Example*

Analytics Use Cases

The final section of the analytics infrastructure model is the use cases built on all this work that you performed: the "analytics solution." Figure 2-15 shows some examples of generalized use cases that are supported with this example. You can build a predictive application for your memory case and use survival analysis techniques to determine which routers will hit this memory leak in the future. You can also use your analytics for decision support to management in order to prioritize activities required to correct the memory issue. Survival analysis here is an example of how to use common industry intuition to develop use cases for your own space. Survival analysis is about recognizing that something will not survive, such as a part in an industrial machine. You can use the very same techniques to recognize that a router will not survive a memory leak.

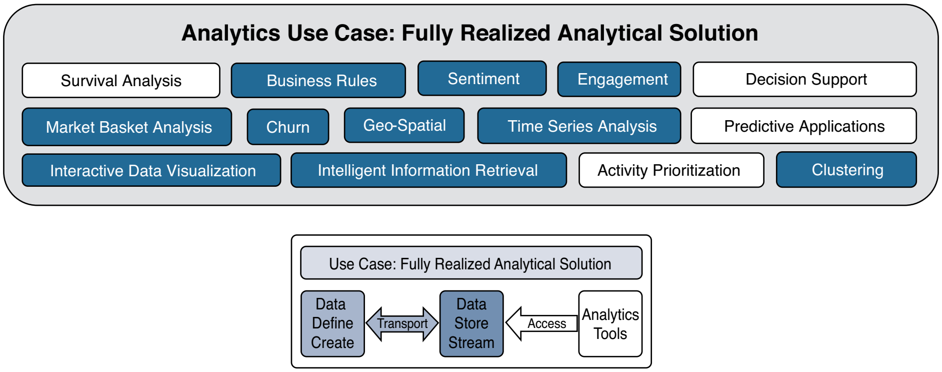


Figure 2-15 *Analytics Infrastructure Model Analytics Use Cases Example*

As you go through the analytics use cases in later chapters, it is up to you and your context bias to determine how far to take each of the use cases. Often simple descriptive analytics or a picture of what is in the environment is enough to provide a solution. Working toward wisdom from the data for predictive, prescriptive, and preemptive analytics solutions is well worth the effort in many cases. The determination of whether it is worth the effort is highly dependent on the capabilities of the systems, people, process, and tools available in your organization (including you).

Figure 2-16 shows where fully automated service assurance is added to the analytics infrastructure model. When you combine the analytics solution with fully automated remediation, you build a full-service assurance layer. Cisco builds full-service assurance layers into many architectures today, in solutions such as Digital Network Architecture (DNA), Application Centric Infrastructure (ACI), Crosswork Network Automation, and more that are coming in the near future. Automation is beyond the scope of this book, but rest assured that your analytics solutions are a valuable source for the automated systems to realize full-service assurance.

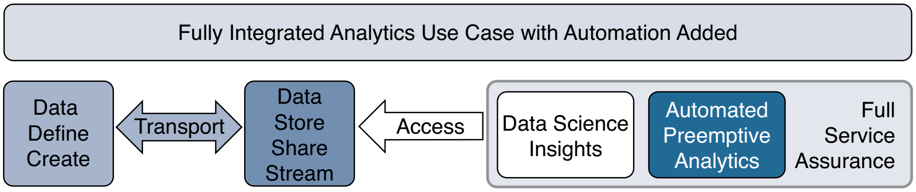


Figure 2-16 *Analytics Infrastructure Model with Service Assurance Attachment*

Summary

Now you understand that there is a method to the analytics madness. You also now know that there are multiple approaches you can take to data science problems. You understand that building a model on captive data in your own machine is an entirely different process from deploying a model in a production environment. You also understand different approaches to the process and that you and your stakeholders may each show preferences for different ones. Whether you are starting with the data exploration or the problem statement, you can find useful and interesting insights.

You may also have had your first introduction to the overlays and underlays concepts, which are important concepts as you go deeper into the data that is available to you from your network in the next chapter. Getting data to and from other overlay applications, as well as to and from other layers of the network is an important part of building complete solutions.

You now have a generalized analytics infrastructure model that helps you understand how the parts of analytics solutions come together to form a use case. Further, you understand that using the analytics infrastructure model allows you to build many different levels of analytics and provides repeatable, reusable components. You can choose how mature you wish your solution to be, based on factors from your own environment. The next few chapters take a deep dive into understanding the networking data from that environment.

This page intentionally left blank

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X-Y-Z