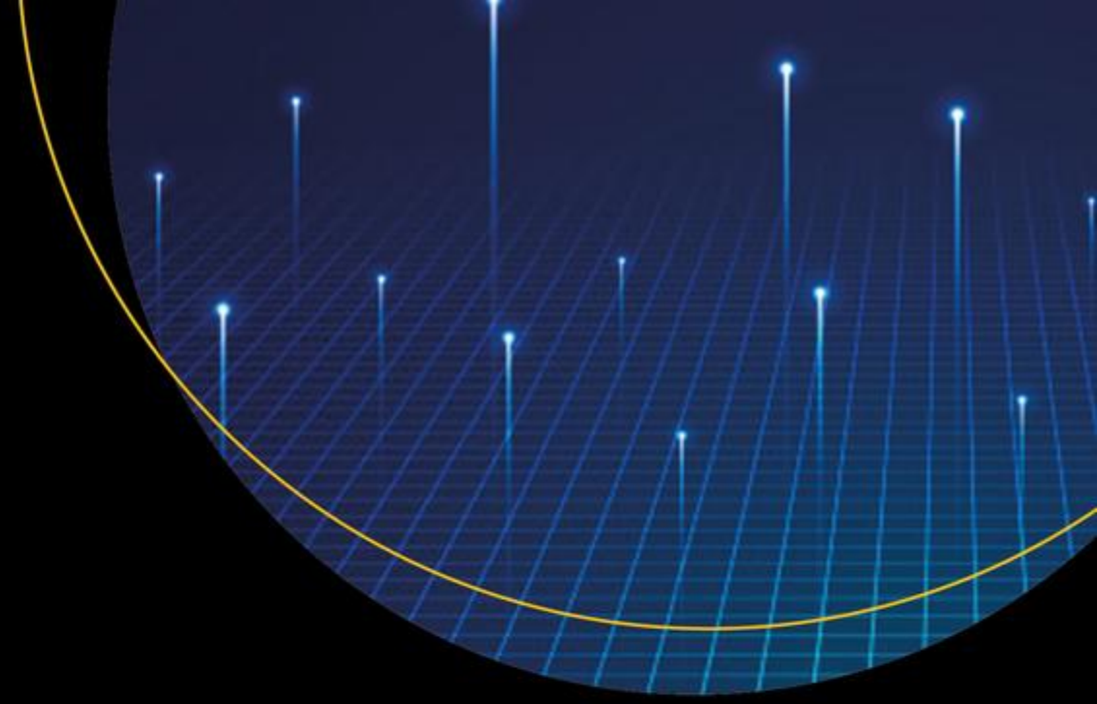


Deploying AI in the Enterprise

IT Approaches for Design, DevOps, Governance, Change Management, Blockchain, and Quantum Computing

—

Eberhard Hechler
Martin Oberhofer
Thomas Schaeck

Foreword by Srinivas Thummalapalli

Apress®

About This Book

Your company has committed to AI. Congratulations, now what? This practical book offers a holistic plan for implementing AI from the perspective of IT and IT operations in the enterprise. You will learn about AI's capabilities, potential, limitations, and challenges. This book teaches you about the role of AI in the context of well-established areas, such as design thinking and DevOps, governance and change management, blockchain, and quantum computing, and discusses the convergence of AI in these key areas of the enterprise.

Deploying AI in the Enterprise provides guidance and methods to effectively deploy and operationalize sustainable AI solutions. You will learn about deployment challenges, such as AI operationalization issues and roadblocks when it comes to turning insight into actionable predictions. You also will learn how to recognize the key components of AI information architecture, and its role in enabling successful and sustainable AI deployments. And you will come away with an understanding of how to effectively leverage AI to augment usage of core information in Master Data Management (MDM) solutions.

What You Will Learn

- Understand the most important AI concepts, including machine learning and deep learning
- Follow best practices and methods to successfully deploy and operationalize AI solutions
- Identify critical components of AI information architecture and the importance of having a plan
- Integrate AI into existing initiatives within an organization
- Recognize current limitations of AI, and how this could impact your business
- Build awareness about important and timely AI research
- Adjust your mindset to consider AI from a holistic standpoint
- Get acquainted with AI opportunities that exist in various industries

Who This Book Is For

IT pros, data scientists, and architects who need to address deployment and operational challenges related to AI and need a comprehensive overview on how AI impacts other business critical areas. It is not an introduction, but is for the reader who is looking for examples on how to leverage data to derive actionable insight and predictions, and needs to understand and factor in the current risks and limitations of AI and what it means in an industry-relevant context.

For more information or to purchase the book, please go to:
https://www.apress.com/gp/book/9781484262054

Deploying AI in the Enterprise

IT Approaches for Design, DevOps, Governance, Change Management, Blockchain, and Quantum Computing

Eberhard Hechler
Martin Oberhofer
Thomas Schaeck

Foreword by Srinivas Thummalapalli

Apress®

Deploying AI in the Enterprise: IT Approaches for Design, DevOps, Governance, Change Management, Blockchain, and Quantum Computing

Eberhard Hechler
Sindelfingen, Germany

Martin Oberhofer
Boeblingen, Germany

Thomas Schaeck
Boeblingen, Germany

ISBN-13 (pbk): 978-1-4842-6205-4
ISBN-13 (electronic): 978-1-4842-6206-1
https://doi.org/10.1007/978-1-4842-6206-1

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Joan Murray
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484262054. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

To my wife, Irina, and our two sons, Lars and Alex, for their continuing support and understanding in writing this book on long evenings and weekends instead of spending time with them.

—Eberhard Hechler

To my wife, Kirsten, and our two sons, Damian and Adrian, thank you for all the love and inspiration you give me every day.

—Martin Oberhofer

To my wife, Annette, and our children, Amelie and Felix, for their support and patience while I was contributing to this book.

—Thomas Schaeck

About the Authors

Eberhard Hechler is an Executive Architect at the IBM Germany R&D Lab. He is a member of the Db2 Analytics Accelerator development group and addresses the broader data and AI on IBM Z scope, including machine learning for z/OS. After 2.5 years at the IBM Kingston Lab in New York, he worked in software development, performance optimization, IT/solution architecture and design, open source (Hadoop and Spark) integration, and master data management (MDM).

He began to work with Db2 for MVS, focusing on testing and performance measurements. He has worked worldwide with IBM clients from various industries on a vast number of topics, such as data and AI including analytics and machine learning, information architectures (IA), and industry solutions. From 2011 to 2014, he was at IBM Singapore, working as Lead Big Data Architect in the Communications Sector of IBM's Software Group.

Eberhard has studied in Germany and France and holds a master's degree (Dipl.-Math.) in pure mathematics and a bachelor's degree (Dipl.-Ing. (FH)) in electrical engineering. He is a member of the IBM Academy of Technology Leadership Team and coauthored the following books: *Enterprise MDM*, *The Art of Enterprise Information Architecture*, and *Beyond Big Data*.

Martin Oberhofer is an IBM Distinguished Engineer and Executive Architect. He is a technologist and engineering leader with deep expertise in master data management, data governance, data integration, metadata and reference data management, artificial intelligence, and machine learning. He has a proven track record of translating customer needs into software solutions, working collaboratively with globally distributed development, design, and offering management teams. He guides development teams using Agile and DevOps software development methods. He can easily adapt to ever-present challenges. Recently, he also started to dig into the blockchain technology space, exploring opportunities to bring analytics capabilities to the blockchain realm.

Previous to his current assignment in the IBM Data and AI development organization, Martin worked with many large clients worldwide at the enterprise level, providing thought leadership on data-centric solutions. In this role, he demonstrated his ability to think horizontally to bring business and IT together by communicating solutions to complex problems in simple terms.

He is an elected member of the IBM Academy of Technology and the TEC CR. He is a certified IBM Master Inventor with over 100 granted patents and numerous publications, including 4 books.

Thomas Schaeck is an IBM Distinguished Engineer (technical executive) at IBM Data and AI, leading Watson Studio on IBM Cloud (Cloud Pak for Data) Desktop and integration with other IBM offerings. Watson Studio is a cloud-native collaborative data science and AI environment for data scientists, data engineers, AI experts, business analysts, and developers, allowing teams to gain insights, train, define and deploy ML/DO models, and get from insights to optimal actions. Previously, Thomas led architecture and technical strategy for IBM Connections, WebSphere Portal, and IBM OpenPages. On a 1-year assignment in the USA in 2013–2014, Thomas led transformation of architecture, technical strategy, and DevOps process for IBM OpenPages Governance Risk Compliance, drove adoption of IBM Design Thinking, and became a trusted partner for major IBM OpenPages customers.

Previously, Thomas led architecture and technical strategy for IBM Connections and integration with WebSphere Portal, enterprise content management, business process management, and design and development of Smart Social Q&A, became a trusted partner for large-enterprise customers as well as customer councils, and helped accelerate sales. On a 2-year assignment in the USA in 2004–2006, Thomas led collaboration software architecture, development, and performance for messaging and web conferencing, achieving acceleration of development productivity and large improvements in performance and scalability.

Thomas also led architecture and technical direction for WebSphere Portal Platform and development of the WebSphere Portal Foundation, initiated and led the portal standards Java Portlet API and OASIS WSRP and Apache open source reference implementations, and initiated and led the Web 2.0 initiative for WebSphere Portal. As a trusted portal architect and leader in portal integration standards, he played a key role in winning the hearts and minds of initial reference customers and then many enterprise customers in Germany and Europe.

About the Technical Reviewer

Mike Sherman has over 35 years of marketing, market research, and CRM/big data experience. He helps clients address marketing opportunities through understanding end users' needs, turning them into insight/data specifications, and converting the output into clear, actionable results.

Mike recently published his first (and last!) book, *52 Things We Wish Someone Had Told Us About Customer Analytics*, coauthored with his son Alex. The book captures real-life lessons learned over their careers, with a focus on practical applications of analytics that connect methodologies and processes to impactful outcomes.

Mike began his career at Procter & Gamble, where he managed both new and established brands. Mike spent 17 years with McKinsey & Company; while there, he created their Asia-Pacific marketing practice and was a founder of their global CRM practice. Mike was also Global Head of Knowledge Management for Synovate, where he led efforts to improve the value clients obtain from research. At SingTel and Hong Kong Telecom, he set up their big data teams and drove the use of both customer data and customer research to help the business understand customer and customer data opportunities.

Mike has been based in Asia since 1997 and has supported work in almost every country in the Asia-Pacific region. Mike has extensive experience in the telecom, retail, financial services, consumer electronics, and FMCG industries.

Mike has an MBA, High Distinction (Baker Scholar) from Harvard Business School, and two bachelor's degrees, magna cum laude, from the Wharton School and College, University of Pennsylvania.

Mike is a frequent speaker at conferences and published several times in the McKinsey Quarterly on marketing issues in developing Asian markets. He is the former Board Chair of AFS-USA, a leading high-school foreign exchange organization, and an avid traveler, having visited over 140 countries.

Foreword

Artificial intelligence is a broad term that has captured people's interest. Until recently, it was confined to scientific circles as one of the most advanced branches of study and struggled to find its way into the industrial arena. While many reasons can be cited for its delayed entry, I strongly believe it is due to the lack of prescriptive books, like the one you hold in your hand. It is the ever-increasing speed of processors churning out data in massive volume that necessitated automation in the data space. AI precisely handles this task of "understanding the data" while producing analytical results. With increasing volumes of data, industries were frantically searching for new tools to analyze the data, and AI came to the rescue at the right time. However, AI is complex enough that only a few in the mainstream can make sense of it. This author team comes to the rescue at the right moment, by providing insight with great examples.

AI has started showing promise recently in the commercial space as many have started using AI for simple use cases to eliminate some mundane tasks that can be automated. From a very early age of my childhood, I was fascinated with the scientific approach to solving problems and the curiosity only grew with time for me. AI grabbed my attention when I was looking for ways to address various problems the financial industry was trying to solve. At that time, I had an opportunity to design systems as a chief enterprise architect, positioning me to merge my curiosity with my role at work. I have been following AI since then and looking for ways to use it in the industrial space. There have been many use cases that came to reality recently using AI. The prompt-based phone answering systems we used to use are now being replaced by speech-enabled systems, empowering customers to directly ask for what they want rather than the system taking them (painfully) through prompts. The Internet is now fully sprinkled with chatbots to take over customer service online. The financial industry has long been using machine learning (a branch of AI) for predictive data analytics and automated decision making in various applications. Tesla and many other firms have started using AI in autonomous vehicles. Increasingly, many companies have started exploring AI/ML for potential use cases in their space.

AI/ML is a complex subject and requires well-versed authors to introduce it to a widespread audience. In this book, Eberhard, Martin, and Thomas did a great job of introducing the subject in plain English. They have successfully bridged the gap between AI and the mainstream audience and explained the subject in simple terms. This book is appropriate for audiences ranging from enthusiastic readers to scientific researchers. It will help people in various roles, such as analysts, programmers, architects, business managers, senior managers, and C-level executives, to be exposed to the subject. The book will bring its audience from a level of having no knowledge about AI/ML to a good understanding of the field, while creating the ability for people to use the tools in their field.

The authors take the reader into the topic of creating an information architecture (IA) surrounding AI/ML. The goal is to create a structure for deploying AI/ML in any organization and helping business executives to absorb it in their own organization. The book very cleanly outlines imperatives for architecture surrounding AI, even to a novice reader. The authors have been able to successfully identify various entities in IA where there is none presently available in the industry in the field of AI. The book will help appropriate personnel in proper positions achieve organizational success.

After IA on AI, the reader is guided through the process of operationalizing AI instead of just being left with an understanding of the subject. Understanding and implementation are two entirely different aspects which the authors clearly understand. The reader is not left with unanswered questions about how to operationalize AI. The authors clearly know that it becomes more relevant in AI than in other fields and have helped to guide the reader in the process. To that end, they have further classified different aspects of AI into subdomains with simple explanations. I find it very important for any level of AI expert to take a concept to implementation.

The authors not only focus on bringing AI/ML into the mainstream but also look at the big picture while correlating AI/ML to other fields such as blockchain and quantum computing. They clearly demonstrate a broad understanding while bringing AI/ML to IT fields such as governance, change management, and DevOps. As AI is a new field and still in experimental stages in the commercial arena, it's not easy to put controls around it. However, the authors understand the importance of governance in the field of IT and don't leave AI without it. They explain the basic need of it in IT before drawing the reader into AI governance itself. The authors have covered the entire scope of AI in language that can be easily understood.

The authors not only highlight AI/ML for their benefits but also show the limitations of the field and suggest new advances required to push the envelope. The book will help professionals prepare for future advances in their field while they are deploying AI at their organization. I personally like this aspect of illustrating the limitations of AI. It shows the depth of an author in a given field to draw out the limitations of the field. Only a well-rounded expert in the field is capable of providing limitations of the field, and the authors no doubt show their depth. I would encourage everyone in IT and scientific fields to read through this book to get an understanding of the field from a fresh set of eyes and to develop a new perspective. The book is a very good read, even for those who are not in technical fields, as it is written in simple English and is in reach to a casual, interested reader of AI. It will improve understanding of the field while enriching one's capability to invite AI/ML into their organization, regardless of the industry.

I congratulate the authors on their well-written book and encourage them to continue to provide valuable bridges and insights in the future.

Srinivas Thummalapalli
Chief Enterprise Architect
Fifth Third Bank
July 2020

Acknowledgments

Writing a book is a much harder endeavor than we thought and more rewarding than we could have imagined. It requires subject-matter expertise and insight, but also motivation and inspiration. Staying engaged, driving the project forward, improving the chapters, making them more readable, and finding new motivation somewhere weren't always so easy. But now it's done.

We are eternally grateful to the many IBM colleagues, domain experts, and leaders we have worked with around the globe. Collaborating with universities provided us with an invaluable and product-agnostic view regarding artificial intelligence (AI) research topics. Numerous enterprises and organizations that we have had an opportunity to work with in the recent years have provided us with the inspiration and insight in elaborating on some of the AI challenges – and coming up with ideas – in deploying AI into the enterprise.

A very special thanks to *Stephane Rodet*, the Lead UX Engineer from the IBM Germany R&D Lab, who has helped us so much in getting the figures of this book into an attractive and consumable form.

Last but not least, thanks to everyone on the Apress team who helped us so much. Special thanks to *Joan Murray*, the ever-patient acquisitions editor, and *Jill Balzano*, our amazing coordinating editor, the greatest cover designer we could ever imagine.

Book Layout

This book is for a reader who is looking for guidance and recommendations on how to overcome AI solution deployment and operationalization challenges in an enterprise and is, furthermore, eagerly interested in getting a comprehensive overview on how AI impacts other areas, such as design thinking, information architecture, DevOps, blockchain, and quantum computing – to name a few. The anticipated reader is looking for examples on how to leverage data to derive to actionable insight and predictions and tries to understand current risks and limitations of AI and what this means in an industry-relevant context. We are aiming at IT and business leaders, IT professionals, data scientists, software architects, and readers who have a general interest in getting a holistic AI understanding.

The chapters of this book are organized into four main parts.

Part I: Getting Started sets the scene for the book in terms of providing a short introductory chapter, an AI evolutionary perspective including technological advancements, and a short description of the most important AI aspects with machine learning (ML) and deep learning (DL) concepts.

It consists of the following three chapters:

- **Chapter 1: AI Introduction** gives an overview of AI in enterprises, providing examples of high-value use cases and showing how AI can be applied in practice. It describes how to increase enterprise automation using AI and introduces the AI life cycle from an enterprise point of view.
- **Chapter 2: AI Historical Perspective** describes why the theoretical AI underpinning of the second half of the twentieth century led to the remarkable AI boost in the last decade. We also venture a glimpse into the future, briefly elaborating on technological advancements that we will most likely observe in the near future.

- **Chapter 3: Key ML, DL, and DO Concepts** introduces key concepts of ML and decision optimization (DO) and explains the differences between these two concepts. We also discuss labeling data in smart ways to minimize labor cost and expert time in labeling and introduce the concept of automated creation of AI models.

Part II: AI Deployment concentrates on successful AI deployments by advocating the implementation of a pervasive information architecture for AI, which is an essential component of every AI deployment that is all too often neglected. We are then providing examples how to turn data into actionable predictions and insight, describing how to leverage ML-based matching for improved and trusted core information management, and sharing guidelines with the reader to overcome operationalization challenges in enterprise environments, including key design thinking and DevOps aspects in the context of AI.

It consists of the following four chapters:

- **Chapter 4: AI Information Architecture** elaborates on the role of an information architecture to deliver a trusted and enterprise-level AI foundation. This chapter is important to the reader in order to fully understand the impact of AI on an existing information architecture to deploy sustainable AI solutions.
- **Chapter 5: From Data to Predictions to Optimal Actions** explains how predictions from ML and decision optimization can be combined to achieve optimal outcomes for enterprises, including a set of practical examples.
- **Chapter 6: The Operationalization of AI** deals with the implementation of AI artifacts into an often highly complex and diverse enterprise environment. This includes real-time scoring; monitoring of, for instance, ML models to maintain their accuracy and precision; and turning data into actionable insight.
- **Chapter 7: Design Thinking and DevOps in the AI Context** describes how design thinking and DevOps methods can be applied to develop AI systems and devices, products and tools, and applications. We also elaborate on how AI and its siblings can be leveraged and infused into design thinking and DevOps concepts.

Part III: AI in Context takes into consideration that AI doesn't stand by itself, it exists within a larger context. This third part describes AI in the context of other key initiatives across industries, such as blockchain, quantum computing, governance and master data management, and change management.

It consists of the following five chapters:

- **Chapter 8: AI and Governance** describes AI and governance aspects and, furthermore, discusses the need for explainability, fairness, and traceability. Since AI-based decision making ought to be meaningful and human comprehensible, AI comes with a new dimension of governance imperatives designed to ensure transparency, trust, and accountability.
- **Chapter 9: Applying AI to Data Governance and MDM** provides a deep dive into applying ML to master data management (MDM) and data governance solutions. It specifically highlights the application of ML to improve required matching algorithms for MDM and to discover hidden relationships in core enterprise information.
- **Chapter 10: AI and Change Management** sheds some light on change management in the context of AI and introduces key aspects of AI change management, such as identifying and analyzing sentiments for a more targeted change management with an optimized outcome.
- **Chapter 11: AI and Blockchain** describes the applicability of AI to blockchain, which by itself is still a relatively new concept, and provides examples, for instance, to increase tamperproof audit trail for AI model versions, data sets used in training, and many others.
- **Chapter 12: AI and Quantum Computing** looks at some AI problems, which are likely to benefit from quantum computing. The promise of quantum computing to surpass "classical" computers for some computational problems may have a profound impact on solving AI problems, for instance, complex back-propagation algorithms used to learn high-dimensional artificial neural networks (ANNs).

Part IV: AI Limitations and Future Challenges discusses current AI limitations and challenges, some of which are subject to research, while others may constitute insoluble challenges that will leave room for human beings to fill that gap – even in the distant future. Some closing remarks and an outlook into the future of AI will conclude this final part of the book.

It consists of the following two chapters:

- **Chapter 13: Limitations of AI** addresses the promise of AI with its breathtaking range of applications that seem to be without limits. And yet, even for AI, there are a number of limits and future challenges, as we learn about in this chapter.
- **Chapter 14: In Summary and Onward** gives an outlook on likely future evolution of AI and AI adoption and shares thoughts on possible consequences.

CHAPTER 5

From Data to Predictions to Optimal Actions

The concept of optimizing decisions based on predictions considering additional data and constraints introduced in Chapter 1, "*AI Introduction*," is often critical to solve real business problems. Decision optimization (DO) takes predictive insight one step further and guarantees that an optimal combination of business-relevant actions can be taken based on predictive insight with relevant context.

In this chapter, we explore this area in more depth and give practical examples of how ML and DO can be combined to get from data to predictions to optimal decisions and resulting actions.

Use Case: A Marketing Campaign

At the beginning of a new year, a bank wants to run a targeted marketing campaign in order to maximize revenue across its banking products and customer base, while avoiding sending customers too many messages.

The bank owns comprehensive customer profile information, data about customers' deposits with the bank, income, age, household size, and data about what investments and products of the bank customers already have as of the past year.

The bank's data steward team curates the data that is deemed necessary for the marketing campaign project and makes it available to a team of data scientists along with responses to prior marketing campaigns, so that they can create ML models to be used in order to optimize revenue and profit. For this campaign, targeting the most promising customers with the most suitable products and services is of essence.

E. Hechler et al., *Deploying AI in the Enterprise*, https://doi.org/10.1007/978-1-4842-6206-1_5

Naïve Solution: ML 101

The team of data scientists builds an ML model for predicting which customers are likely interested in what products or services in the new year.

The predictive model is trained using input data composed of relevant features derived from customer profile information, customer's cash deposits, the products and services they already have in their portfolio, and the products they bought in the past year.

The data scientists provide the ML model to the IT department, with IT team members to deploy and to run a batch scoring job, computing daily for all customers of the bank which products and services they are most likely to buy.

The result is a table in a database with a row for each of the tens of thousands of customers of the bank, including the predicted product each customer individually would be most likely to buy.

These predictions including the probability of buying could be used to send each customer a marketing message, offering them the product that they are most likely to buy; however, in doing so, the bank might run into difficulties:

- **Budget constraints**: The budget for the marketing campaign might be exceeded. The campaign may not be sufficiently targeted, meaning that too many clients may receive an offer that they simply reject. As a result, too much money would be spent without getting an adequate return for it.
- **Product and service availability**: The bank might run out of some products it only has limited financial backing for or may face constraint in being able to provide a time- and resource-intensive service to only a handful of selected clients. If too many clients are offered to buy such products, the bank may not be able to fulfill what they offered with defined quality and services standards.
- **Risks, limitations, and costs**: Product and service offers with associated risks, limitations, and cost require careful additional consideration to avoid negative effects on profitability or risk taken on by the bank.
- **Risk of customer negativity***:* Product offers that are perceived as "creepy," giving the customer a sense that the bank knows and uses too much data about them, can cause anger and result in negative perception of the bank.

In practice, a brute-force approach to send every customer an offer for the product predicted to be what they are most likely to buy, disconnected from context and real-world constraints, would be suboptimal. It could cause high cost, risk, and potentially customer dissatisfaction if making too many offers the bank could in the end not all fulfill.

Refined Solution: ML plus DO

The preceding naïve solution falls short because it makes decisions and takes actions without considering context and constraints. To act on each prediction could make sense if the number of predictions would be much smaller than available resources. But as soon as predictions would lead to a nontrivial number of decisions and resulting actions, it becomes important to consider the whole set of possible decisions in the context of related data and constraints. Driving to the most optimal relevant and context-aware decisions is what decision optimization (DO) solves.

With DO, data scientists or optimization experts can define an *optimization problem* consisting of a set of constraints that need to be honored, objective(s) to be optimized, and data to be considered in solving the problem. The problem defined this way is then solved using a DO engine, generating an *optimized* set of decisions as a result.

Applying DO to the bank's marketing campaign project ensures that within a given marketing budget and product quantities, the optimal set of product offers is made to the right customers based on the predictions from the predictive ML model, optimizing revenue and preventing offers that the bank might not be able to fulfill.

Example: ML plus DO

In this section, we show you an example how ML and DO can be used together in IBM Watson Studio[1], by creating a predictive ML model and a prescriptive DO model in a project, deploying both artifacts and making them available for use by a process that can invoke the predictive ML model followed by solving the prescriptive DO model[2].

[1]See [1] for more information on IBM Watson Studio.

[2]We are using IBM Watson Studio for this example; however, similar tools from a variety of different vendors could be used as well.

Create a Project

A project owner creates a project and can add various personas, such as data scientists, subject matter experts, and optimization experts as needed to work on the project as a team. Only project members can access the secure environment provided by the project.

As you can see in Figure 5-1, within the scope of a project, these different personas efficiently collaborate with each other to perform various tasks, such as connecting to data sets and refining them. Furthermore, various notebooks can be used for the data to be explored, analyzed, and visualized. Under the umbrella of the defined project, ML and DL models as well as DO models can be developed, trained, validated, deployed, and finally tested.

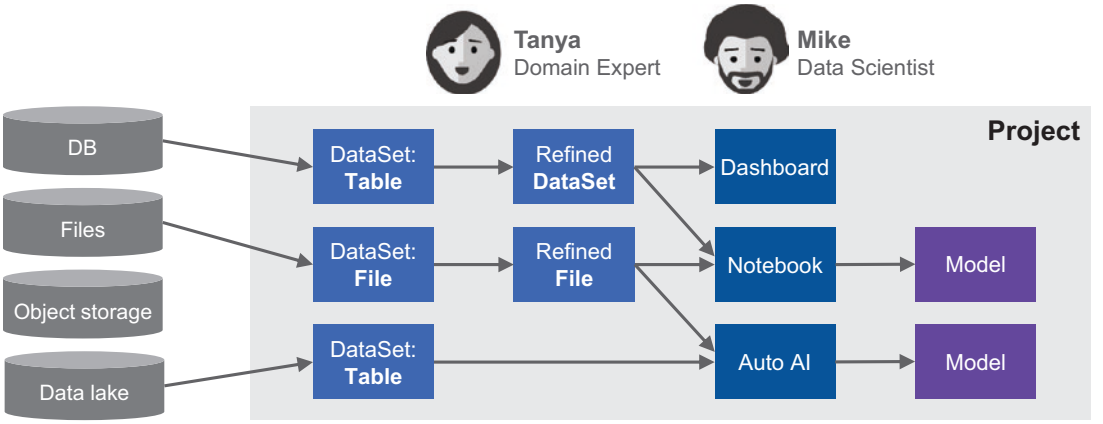


Figure 5-1. *Creating a Data Science Project*

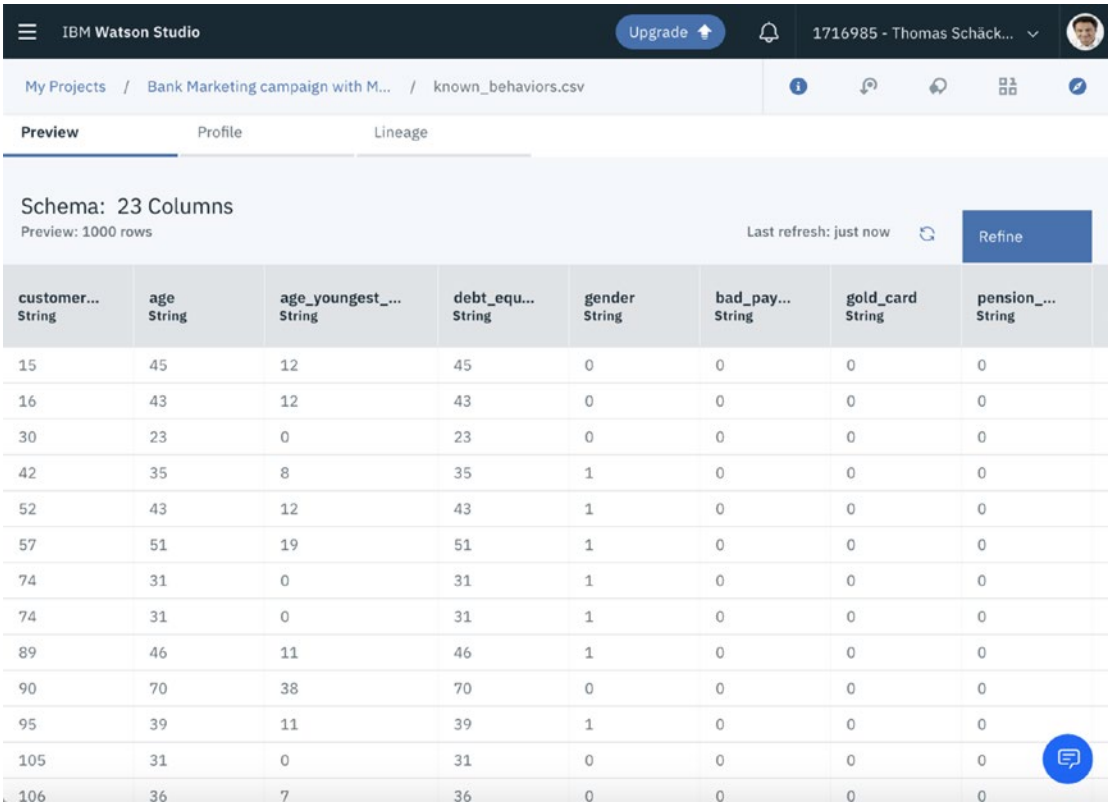
In the following sections, we describe these various tasks in more detail and visualize some of these steps using IBM Watson Studio.

Connect Data

Project members can connect to data in databases, object stores, or other data sources via connections and reference or copy subsets of relevant data as data assets in the project. If the original data contains sensitive information that is not intended to be used in the project, the person who adds it to the project can omit certain columns when adding the data to the project.

In this case, a data steward was added to the project to make the required data available for use in the project. The data steward selects the data that data scientists, data engineers, and other personas are allowed to use for the project and adds it to the project as a database table or CSV file. Data from various data source systems can be provisioned.

The objective is to make available all data that is relevant for this particular business solution. This sounds like a rather trivial task, which is often grounded on the assumption that relevant data is simply available and easily accessible. Nevertheless, making relevant data available for data engineers and data scientists can be a challenging endeavor.



customer... String	age String	age_youngest_... String	debt_equ... String	gender String	bad_pay... String	gold_card String	pension_... String
15	45	12	45	0	0	0	0
16	43	12	43	0	0	0	0
30	23	0	23	0	0	0	0
42	35	8	35	1	0	0	0
52	43	12	43	1	0	0	0
57	51	19	51	1	0	0	0
74	31	0	31	1	0	0	0
74	31	0	31	1	0	0	0
89	46	11	46	1	0	0	0
90	70	38	70	0	0	0	0
95	39	11	39	1	0	0	0
105	31	0	31	0	0	0	0
106	36	7	36	0	0	0	0

Figure 5-2. *Data Sets Associated with a Project*

The data sets include relevant columns from the bank's customer records, as illustrated in Figure 5-2 that shows the data set in the data preview in the project.

Refine, Visualize, Analyze Data

Project members can explore and refine data as needed in order to achieve the required level of data quality and representative data distribution. Data engineers are typically in the lead to prepare and transform the available data into a format that is suitable for further consumption by data scientists. Nevertheless, the data exploration, refinement,

and visualization is typically a rather collaborative undertaking, where data scientists work in concert with data engineers. Business domain experts, such as the marketing campaign experts, need to provide guidance to guarantee the required business outcome. Some data may not be useful to make predictions for the future, but still should be included to enable other insights, for example, branch codes and indication of pre-/post-merger records could enable insights such as how these dimensions correlate with performance.

The data refinement can, for instance, be done using the refinery function in the context of the project. It can also be done via coding to refine the data in notebooks[3] using Python, R, or many other languages.

They can be used to interactively visualize and analyze data to better understand correlation and coherence of various data segments. The overall goal is to explore and refine the data, in order to determine relevant features for the creation and training of models.

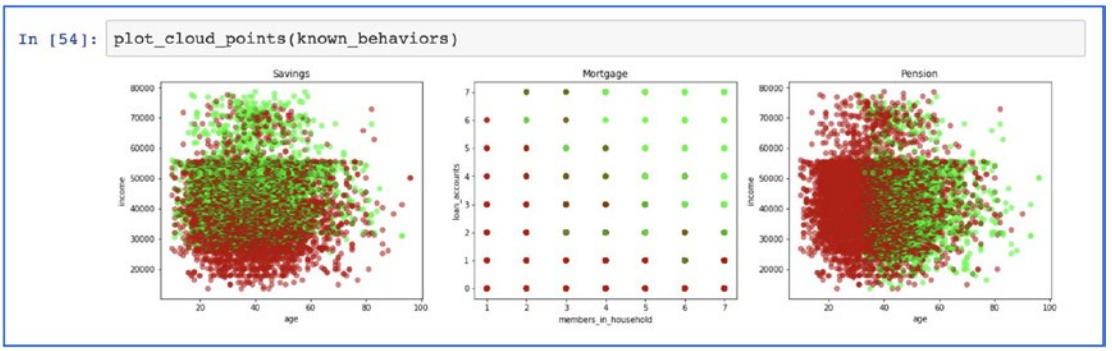


Figure 5-3. *Data Visualization in a Notebook*

As you can see in Figure 5-3, in this example, the data scientists use visualizations directly in a notebook to visualize the data regarding customer purchases from the previous year. Many visualization graphics are possible to use, for instance, discovering important correlation between data columns, KPIs, or defined measures.

This allows a data scientist to gain relevant insight of the data segments, which should then be taken into consideration when developing an ML or DL model.

The following is some insight that can be seen from the visualization in Figure 5-3:

- **Customer income**: The greater a customer's income, the more likely it is that they own a savings account. As we have already pointed out in Chapter 1, "*AI Introduction*," this finding by itself may not be

[3]See [2] and [3] for more information on notebooks for data scientists.

sufficient. In other words, it should be correlated with other insights from this particular customer's transactional records or other customer profile information.

- **Customer age**: The older a customer is, the more likely it is that they own a pension account. This finding may be statistically correct and could very well be used in the development of a ML model; however, it needs to be correlated with other insights for a particular client. For instance, a particular client may already have a pension account.
- **Correlation discovery**:[4] There is a correlation between the number of residents in a customer's household, the number of loan accounts held by the customer, and the likelihood a customer buys a mortgage account, as can be seen in the upper right and lower left corners of the mortgage chart in Figure 5-3.

The preceding examples also emphasize nicely that possible bias should be taken into consideration. The insight could, for instance, be biased toward certain demographic measures (e.g., age, gender, marital status, etc.), which may or may not reflect the reality.

Create and Train Predictive Models

Once the data is in the right shape and well understood, data scientists can create and train predictive machine learning models using integrated tools, for example, using Auto AI, Python Notebooks, or SPSS flows.

We provide some more details of these integrated tools, to give you a better understanding about their capabilities.

Auto AI

Auto AI provides an easy way to create a set of model pipeline candidates by providing a data set and letting Auto AI[5] perform model selection, feature engineering, hyperparameter optimization, and others for a set of pipeline candidates. Data scientists can then explore various metrics of the resulting models, pick the models they like best, and save them to the project.

[4]Correlation discovery is an essential aspect of ML; see [4] for more information on correlation discovery and its applicability in ML.

[5]See [5] for more information on Auto AI.

The resulting models can be evaluated regarding their accuracy and precision by comparing the areas under the ROC and PR curves. This allows a consistent comparison of all models and picking the best model to use.

Auto AI is a great example of how you can get started with AI and ML projects, without necessarily being a data science subject matter expert and skilled in mathematical and statistical methods. It hides some of the complexity that is typically associated with data science tasks.

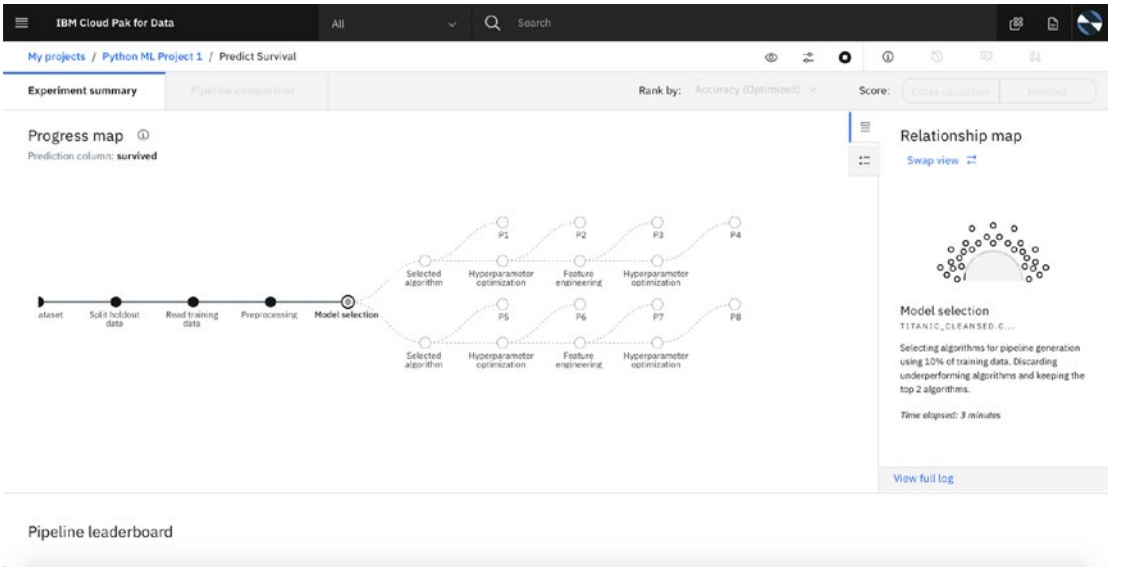


Figure 5-4. *Auto AI Experiment Pipeline Generation*

Figure 5-4 shows a summary of the Auto AI pipeline generation with various steps, such as selecting the algorithms, performing hyperparameter optimization, and feature engineering tasks.

Some of these steps can be performed iteratively to increase the accuracy and precision of the models. As Auto AI executes the steps, it shows progress to the user. When larger data sets are used, users can also sign off and return to their experiment later.

Figure 5-5 shows an overview of all the selected algorithms, candidate training pipelines created based on these algorithms, and feature transformers used by these pipelines after an Auto AI experiment concluded. Apart from picking the model that best fits your needs and saving it, you can also pick a model pipeline and can save it as a Python Notebook.

This allows you to further improve and customize various tasks (e.g., data preparation, feature engineering, hyperparameter settings) and further optimize the accuracy and precision of your ML models.

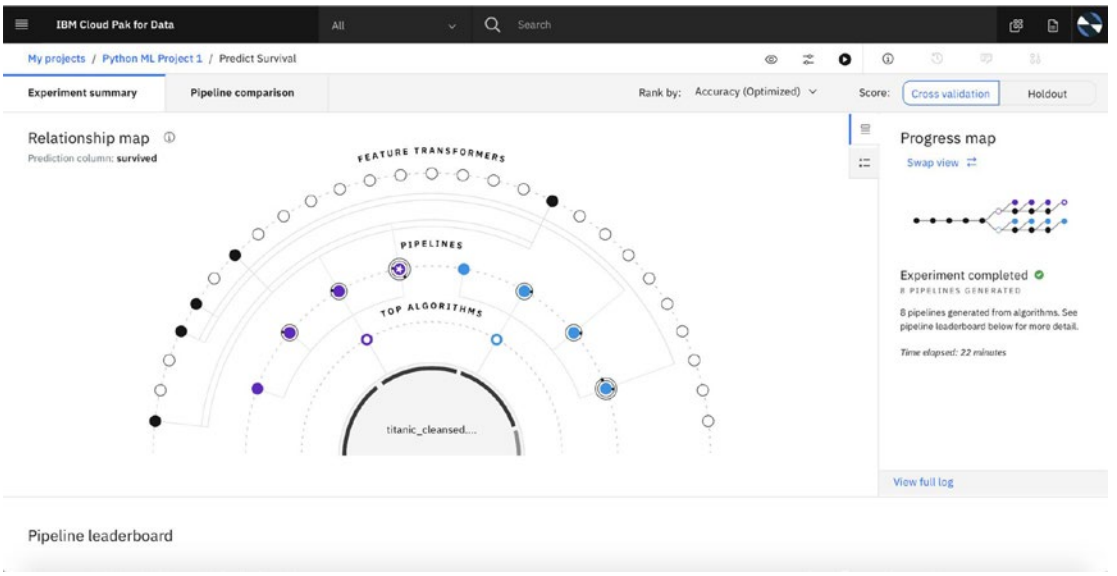


Figure 5-5. *Auto AI Experiment Summary*

You can also optionally interact with Auto AI to specify your own preferences in the automated Auto AI process. The following are some examples:

- Data preparation and advanced data refinery tasks
- Feature engineering, including feature transformations
- Auto AI pipeline optimization
- Hyperparameter optimization (HPO)[6]
- Explainability, debiasing, and fairness
- AI life cycle management to monitor post-deployment performance

[6]See [6] for more information on hyperparameter optimization.

SPSS Flows

SPSS flows allow multiple personas – including business domain experts without coding skills – to create and train models by defining model training flows in a visual editor and running these flows to create, train, and save models to the project.

Figure 5-6 is an illustration of a sample SPSS flow, which can be assembled with no programming skills required.

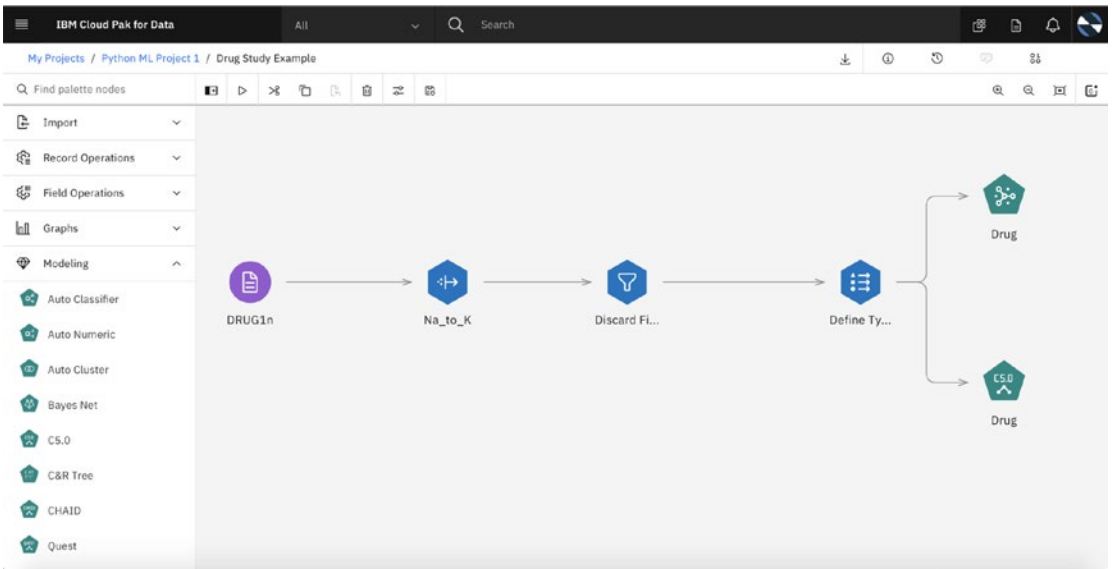


Figure 5-6. *Sample SPSS Flow*

SPSS[7] is an alternative for personas with limited or none programming skills.

Notebooks

As we have mentioned before, notebooks can be used alternatively by data scientists to train models using Python or many other languages like R or Scala, where the trained models can then be saved to the project using the project API from their code in the notebook. Notebooks require programming skills; however, they allow for greatest flexibility and need to be considered as state-of-the-art technique for any data scientist.

[7]See [7] for more information on SPSS.

Python is what we observe to be the most popular programming language among data scientists. Python is also a very popular programming language in general. This has led to a wide range of readily available libraries that can be used from Python code, including a large number of powerful data science, ML, DL, DO, and visualization libraries that make data scientists highly productive.

Figure 5-7 is a snapshot taken while working in a project with a Jupyter Notebook opened in the JupyterLab user interface. It shows the typical combination of displaying code cells and resulting output in a notebook document, which enables literate programming. A notebook is self-documenting; after running, it contains code and resulting insights together with inline text explaining it all. As you can also see in Figure 5-7, JupyterLab can integrate with a Git repository for code management, in this case with a Git repository that was associated with the project.

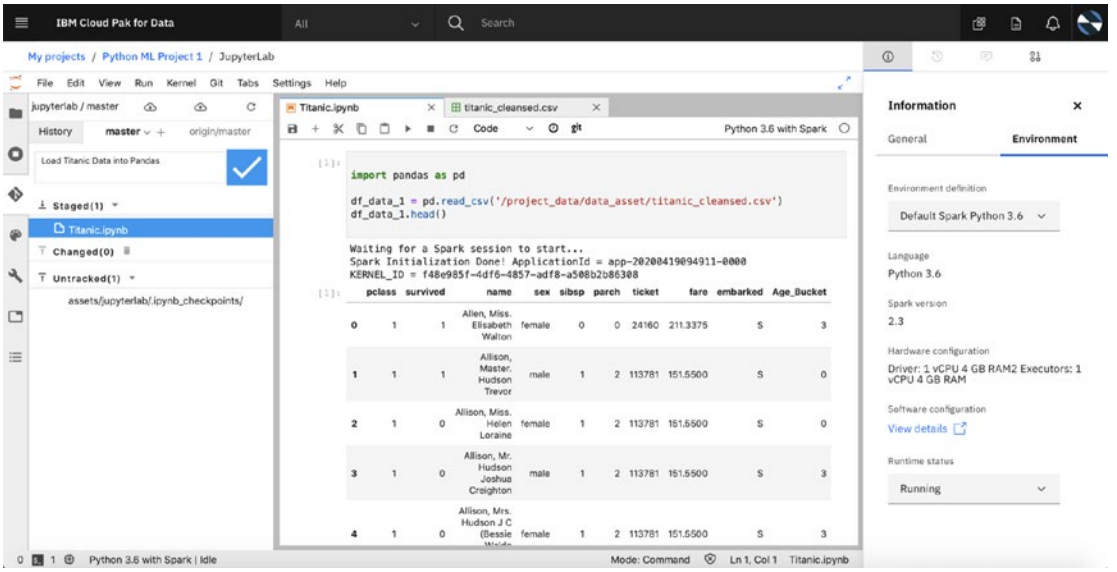


Figure 5-7. *Jupyter Notebook with Python*

In order to run code cells, notebooks need an underlying runtime environment, which in this case is a Python + Spark environment, allowing both single node execution of Python code as well as parallel execution of Python code leveraging the Apache Spark framework.

```
In [18]: from sklearn import svm
         from sklearn import ensemble

In [19]: classifiers = []
         for i,p in enumerate(products):
             clf = ensemble.GradientBoostingClassifier()
             clf.fit(X, ys[i])
             classifiers.append(clf)
```

Figure 5-8. *Python Notebook to Train a Model*

An excellent example illustrating the combination of ML and DO was created by Alain Chabrier[8] from IBM, thought leader and expert in DO and in combining it intelligently with ML. As part of his work in this field, he created a notebook using a banking scenario as an example, from which we included excerpts in this chapter. To generate predictions of customer interest in products, the first step is to train a model using data of customer purchases and customer profile data from the past year.

As can be seen in Figure 5-8, *scikit-learn svm* and *ensemble* are imported to be used in the notebook, and a gradient boosting classifier is used as the underlying ML algorithm. The gradient boosting algorithm is a very popular one for regression and classification problems. Subsequently, they use the model to predict which customers will most likely buy which particular offer in the coming year.

```
In [24]: import warnings
         warnings.filterwarnings('ignore')

In [25]: predicted = [classifiers[i].predict(to_predict) for i in range(len(products))]
         for i,p in enumerate(products):
             to_predict[p] = predicted[i]
         to_predict["id"] = unknown_behaviors["customer_id"]
```

Figure 5-9. *Using the Model for Scoring*

As you can see in Figure 5-9, we are taking each product (savings, mortgage, and pension) and relate this to the chosen customer characteristics, such as age, income, number of members in a household, and number of accounts.

[8]See [8] for more information on DO from Alain Chabrier.

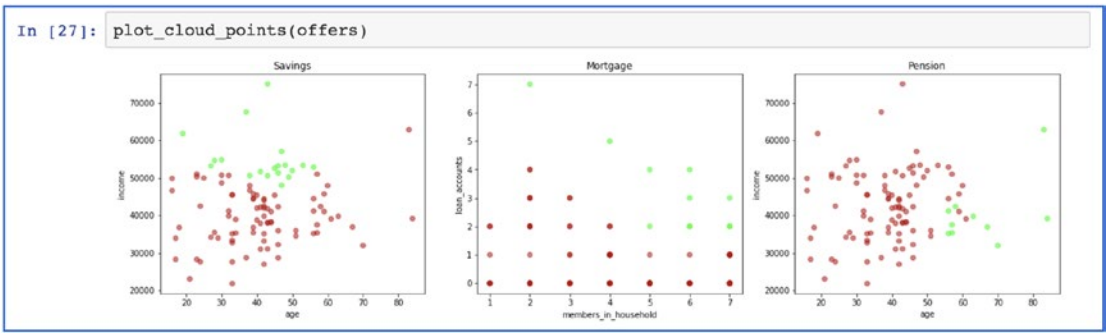


Figure 5-10. *Visualization of Predicted Offers*

The predictions for what to offer can be visualized in the same way as previously the data from the past year was displayed. As you can see in Figure 5-10, we receive a predictive outcome that suggests the following for the individual products:

- **Savings**: There is a higher likelihood for customers with greater income above US $ 50.000 p.a. in need of a savings account.
- **Mortgage**: There is a correlation between the number of members in a household and the number of loan accounts. Households with more than five members and a minimum of two loan accounts are more likely to need a mortgage.
- **Pension**: There is a higher likelihood for customers with an age of greater than 56 and an income of US $ 42.00 or lower to need a pension product.

This example is an illustration; for a realistic scenario, additional product and customer characteristics need to be taken into consideration as well.

Deploy ML Models

Models can be promoted from a project to a *space*, where authorized users can create *model deployments* to serve the models for online or batch scoring.

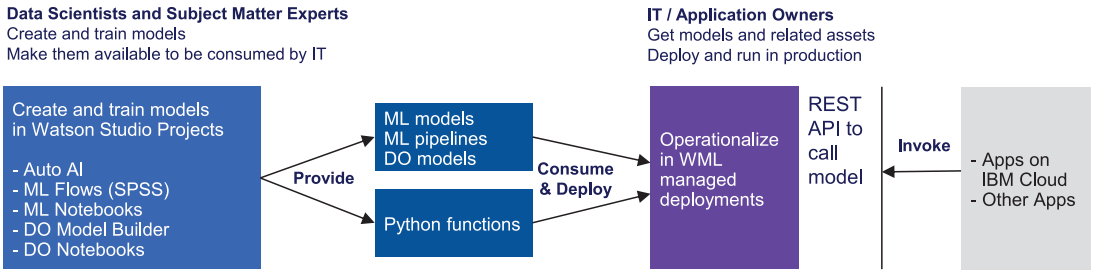


Figure 5-11. *Deploying ML Models*

This makes the predictive ML or DL models accessible through public REST APIs. Applications or business processes can invoke the models through these REST APIs to get predictions. Figure 5-11 depicts the deployment of ML or DL models. Once models are deployed, payload input and output logs with model input data and model prediction output can be recorded in a database table, which can be continuously monitored and analyzed for fairness. This allows to automatically detect poor performance, drift, or bias in model scoring and enables taking corrective action if needed.

Subject matter experts create, train, and validate models and make them available to be operationalized within the IT and business environment. The AI artifacts could be ML or DL models and pipelines, DO models, and Python functions to combine models. IT and applications need to integrate these AI artifacts to be called via REST APIs for scoring.

Create DO Models

As depicted in Figure 5-12, in order to progress from predictions to optimal actions, the team needs to combine ML with DO, so that predictions from an ML model plus other input data can feed into a prescriptive DO model to finally determine optimal actions based on that input.

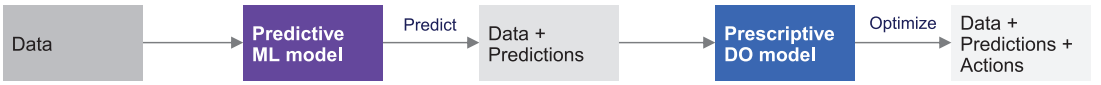


Figure 5-12. *Creating DO Models*

An optimization expert or data scientist familiar with optimization creates and tests a DO model in a notebook or in the DO Model Builder, using data from the project and predictions created by the predictive ML or DL model, to solve for optimal decisions and resulting actions. DO in IBM Watson Studio leverages the advanced docplex engine[9] in order to solve optimization problems.

In this example, being familiar with Python, the data science team chooses to create a Python Notebook and then define and test the DO model in Python code in the notebook. While the docplex Python library is already preinstalled in Python environments in IBM Watson Studio, it can also easily be added in any Python Notebook elsewhere using `!pip install docplex`, providing you with the greatest flexibility. Then a DO model can be created using the following code in a Python Notebook cell:

```
from docplex.mp.model import Model
mdl = Model(name="marketing_campaign")
```

After this step, the decision variables need to be defined, followed by defining the constraints that need to be considered. In this example, the following constraints are relevant:

- Offer only one product per customer.
- Compute the budget and set a maximum on it.
- Compute the number of offers to be made.
- Ensure at least 10% of offers are made via each channel.

```
In [34]: # At most 1 product is offered to each customer
         mdl.add_constraints( mdl.sum(channelVars[o,p,c] for p in productsR for c in channelsR) <=1
                              for o in offersR)

         # Do not exceed the budget
         mdl.add_constraint( mdl.sum(channelVars[o,p,c]*channels.get_value(index=c, col="cost")
                                                 for o in offersR
                                                 for p in productsR
                                                 for c in channelsR)  <= availableBudget, "budget")

         # At least 10% offers per channel
         for c in channelsR:
             mdl.add_constraint(mdl.sum(channelVars[o,p,c] for p in productsR for o in offersR) >= len(
         offers) // 10)

         mdl.print_information()

         Model: marketing_campaign
          - number of variables: 900
            - binary=900, integer=0, continuous=0
          - number of constraints: 104
            - linear=104
          - parameters: defaults
```

Figure 5-13. *Defining Constraints for a DO Model*

[9]See [9] for more information on the docplex engine.

These constraints are defined for the model using `mdl.add_constraint()`, as can be seen in Figure 5-13.

Express the objective

You want to maximize expected revenue, so you take into account the predicted behavior of each customer for each product.

```
In [35]: obj = 0

for c in channelsR:
    for p in productsR:
        product=products[p]
        coef = channels.get_value(index=c, col="factor") * value_per_product[product]
        obj += mdl.sum(channelVars[o,p,c] * coef* offers.get_value(index=o, col=product) for o
in offersR)

mdl.maximize(obj)
```

Figure 5-14. *Defining the Objective*

Then the objective is defined, in this case to maximize revenue using `mdl.maximize()`, as can be seen in Figure 5-14.

Finally, the docplex engine[10] is run to solve the model using `mdl.solve()`, as can be seen in Figure 5-15.

```
In [38]: s = mdl.solve()
         assert s, "No Solution !!!"

In [39]: print(mdl.get_solve_status())
         print(mdl.get_solve_details())

         JobSolveStatus.OPTIMAL_SOLUTION
         status  = integer optimal solution
         time    = 0.0108402 s.
         problem = MILP
         gap     = 0%
```

Figure 5-15. *Running the Docplex Engine*

The output of solving the model is a data frame with the detailed optimized decisions for what offers to make to which customers via which channel.

[10]More information on how to use docplex in notebooks can be found here: https://github.com/IBMDecisionOptimization/docplex-examples.

Deploy DO Models

Like predictive ML models, prescriptive DO models can be deployed to solve optimization problems. This makes the DO models accessible through the Watson ML public REST APIs for access by applications or business processes. Subsequently, with relevant data and ML predictions as input, the docplex engine will solve the problem and generate optimal actions.

This represents a holistic solution in the very specific context of the process or business application.

By deploying both ML and DO models in combination with the same Watson ML service, it becomes easy for applications or processes to call ML models to generate predictions based on relevant data and to then call DO models with data predictions to determine the optimal actions to take.

In this example, the predictions and the optimized decisions are persisted to database tables resulting from batch scoring deployments of the ML and DO models.

Taking ML and DO Models to Production

As we observed in the first chapter, in order to deploy models for production use, they often need to be added and deployed to completely separate production deployment systems.

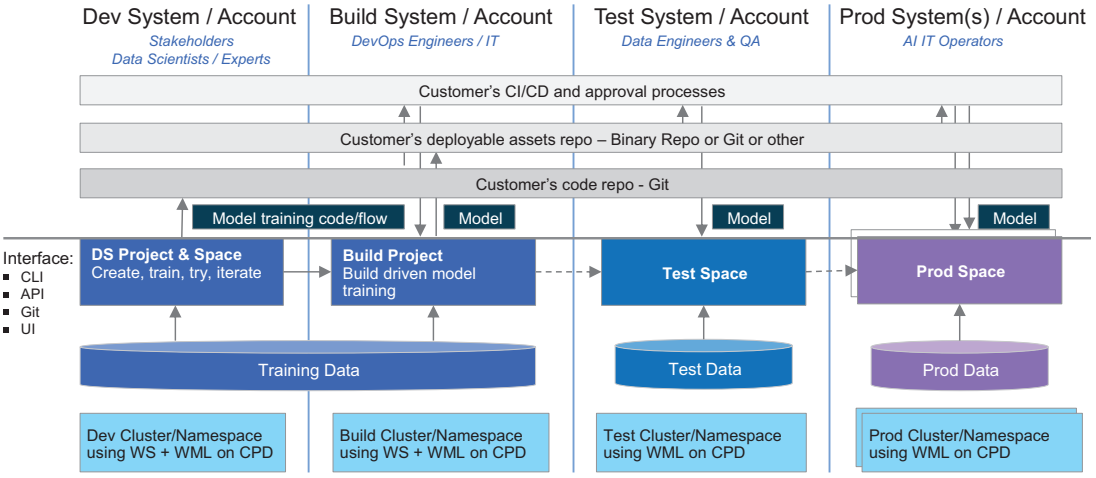


Figure 5-16. *Heterogeneous Development and Production Environment*

To achieve this with Watson Studio, it is possible to, for example, install one OpenShift cluster with IBM Cloud Pak for Data with Watson Studio and Watson Machine Learning as a working environment for data scientists and to install additional clusters with Watson Machine Learning for build, test, and production environments, as depicted in Figure 5-16.

Data scientists can submit their work results to a Git repository or export the assets as a ZIP file, in order to make the ML and DO model assets available to a CI/CD[11] pipeline created and operated typically by a separate IT team. The CI/CD pipeline can then propagate assets to build, test, and eventually production environments, to establish a well-defined process for taking ML and DO model assets to production after reproducible training and testing.

Embedding AI in Applications and Processes

Having deployed ML and DO models for production use, business processes and applications can now use these models to make predictions and determine optimal decisions to drive optimal automated or AI augmented actions.

In our example, the bank's marketing processes can obtain the resulting optimized decisions from a database table that contains the optimal set of decisions for what offers to make to which customers through which channel in order to achieve the optimal result within the available budget and product availability and other constraints.

Key Takeaways

We conclude this chapter with a few key takeaways, summarized in Table 5-1.

[11]CI/CD stands for continuous integration and continuous delivery.

Table 5-1. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	Getting from data to predictions is often not enough	Predictions alone are typically not actionable; it is key to get from predictions to optimal decisions in order to inform the best actions
2	Naïve approach for a marketing campaign	For example, train a model to predict which customers are likely to buy what products – but in practice may not be feasible to market to all potential buyers
3	Smarter approach for a marketing campaign: consider constraints	Take constraints into account, such as budget, availability of product, mailing limitations, and so on and optimize toward well-defined targets
4	Decision optimization for automated solving of optimization problems	With decision optimization, data scientists or optimization experts can define an *optimization problem* consisting of a set of constraints that need to be honored, objective(s) to be optimized, and data to be considered in solving the problem
5	IBM Watson Studio is an example of an environment combining ML + DO	Watson Studio allows to create projects, add members, connect and add data, build machine learning models and decision optimization models, and deploy ML and DO models together for use by applications and processes
6	Decision optimization models can be created via UI or Python	DO designer allows to define models visually; alternatively, DO models can be defined and solved in Jupyter using Python
7	Use DevOps with CI/CD for taking ML and DO models to production	Ensure all relevant artifacts are managed in a trusted code and asset repository (e.g., Git) and can be deployed to test and production systems via automated CI/CD pipelines in a reproducible fashion
8	Achieving HA and DR for model deployments	Deploy models to at least two independent sites with load balancing of requests and with failover if one site fails to serve business critical applications and processes

References

[1] IBM, *IBM Watson Studio*, www.ibm.com/cloud/watson-studio (accessed April 27, 2020).

[2] Galea, A. *Beginning Data Science with Python and Jupyter: Use powerful tools to unlock actionable insights from data*. ISBN-13: 978-1789532029, Packt Publishing, 2018.

[3] Nelli, F. *Python Data Analytics: With Pandas, NumPy, and Matplotlib*. 2nd ed. Edition. ISBN-13: 978-1484239124, Apress, 2018.

[4] Mirkin, B. *Core Data Analysis: Summarization, Correlation, and Visualization* (Undergraduate Topics in Computer Science) 2nd ed. 2019 Edition. ISBN-13: 978-3030002701, Springer, 2019.

[5] Malaika, S., Wang, D. IBM. Artificial Intelligence. *AutoAI: Humans and machines better together*, https://developer.ibm.com/technologies/artificial-intelligence/articles/autoai-humans-and-machines-better-together/ (accessed April 27, 2020).

[6] Naya, G. Towards Data Science. *Available hyperparameter-optimization techniques*, https://towardsdatascience.com/available-hyperparameter-optimization-techniques-dc60fb836264 (accessed April 27, 2020).

[7] IBM. *IBM SPSS software*, www.ibm.com/analytics/spss-statistics-software (accessed April 27, 2020).

[8] Chabrier, A. *Decision Optimization for Data Science Experience: Why?* https://developer.ibm.com/docloud/blog/2018/05/07/do4dsx-why/ (accessed April 27, 2020).

[9] IBM, *The IBM Decision Optimization CPLEX Modeling for Python*, https://pypi.org/project/docplex/ (accessed April 20, 2020).