Practical Data Science

A Guide to Building the Technology Stack for Turning Data Lakes into Business Assets

Andreas François Vermeulen

Apress®

Practical Data Science

A Guide to Building the Technology Stack for Turning Data Lakes into Business Assets

Andreas François Vermeulen

Apress®

Practical Data Science: A Guide to Building the Technology Stack for Turning Data Lakes into Business Assets

Andreas François Vermeulen
West Kilbride North Ayrshire, United Kingdom

ISBN-13 (pbk): 978-1-4842-3053-4 ISBN-13 (electronic): 978-1-4842-3054-1
https://doi.org/10.1007/978-1-4842-3054-1

Library of Congress Control Number: 2018934681

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the author nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Susan McDermott
Development Editor: Laura Berendson
Coordinating Editor: Rita Fernando

Cover designed by eStudioCalamar

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 rights@apress.com, or visit www.apress.com/rights-permissions.

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 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/9781484230534. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Author

Andreas François Vermeulen is a consulting manager for decision science, data science, data engineering, machine learning, robotics, artificial intelligence, computational analytics and business intelligence at Sopra-Steria and a doctoral researcher at the University of St Andrews, Scotland, on future concepts in massive distributed computing, mechatronics, big data, business intelligence, data science, data engineering, and deep learning. He owns and incubates the Rapid Information Factory data processing framework. He is active in developing next-generation processing frameworks and mechatronics engineering, with more than 37 years of international experience in data processing, software development, and system architecture. Andreas is a data scientist, doctoral trainer, corporate consultant, principal systems architect, and speaker/author/columnist on data science, distributed computing, big data, business intelligence, deep learning, and constraint programming. Andreas holds a bachelor's degree from North-West University, Potchefstroom, South Africa; a master of business administration degree from the University of Manchester, England; a master of business intelligence and data science degree from the University of Dundee, Scotland; and Ph.D. from the University of St Andrews.

About the Technical Reviewer

Chris Hillman is a principal data scientist working as part of an international team. With more than 20 years of experience in the analytics industry, Chris has works in various sectors, including life sciences, manufacturing, retail, and telecommunication. Using the latest technology, he specializes in producing actionable insights from large-scale analytical problems on parallel clusters. He has presented at conferences such as Strata, Hadoop world, and the IEEE big data streaming special interest group. Chris is currently studying for a Ph.D. in data science at the University of Dundee, applying big data analytics to the data produced from experimentation into the human proteome, and has published several research papers.

Acknowledgments

To Denise: I am fortunate enough to have created a way of life I love . . . But you have given me the courage and determination to live it! Thanks for the time and patience to complete the book and numerous other mad projects.

To Laurence: Thank you for all the knowledge shared on accounting and finance.

To Chris: thank you. Your wisdom and insight made this great! Best of luck with your future.

To the staff at Apress: your skills transformed an idea into a book. Well done!

Introduction

People are talking about data lakes daily now. I consult on a regular basis with organizations on how to develop their data lake and data science strategy to serve their evolving and ever-changing business strategies. This requirement for agile and cost-effective information management is high on the priority list of senior managers worldwide.

It is a fact that many of the unknown insights are captured and stored in a massive pool of unprocessed data in the enterprise. These data lakes have major implications for the future of the business world. It is projected that combined data scientists worldwide will have to handle 40 zettabytes of data by 2020, an increase of 300 times since 2005.

There are numerous data sources that still must be converted into actionable business knowledge. This achievement will safeguard the future of the business that can achieve it.

The world's data producers are generating two-and-a-half quintillion bytes of new data every day. The addition of internet of things will cause this volume to be substantially higher. Data scientists and engineers are falling behind on an immense responsibility.

By reading this introduction, you are already an innovative person who wants to understand this advanced data structure that one and all now desire to tame.

To tame your data lake, you will require practical data science.

I propose to teach you how to tame this beast. I am familiar with the skills it takes to achieve this goal. I will guide you with the sole purpose of you learning and expanding while mastering the practical guidance in this book.

I will chaper one you from the data lake to the final visualization and storytelling.

You will understand what is in your business's data lake and how to apply data science to it.

Think of the process as comparable to a natural lake. It is vital to establish a sequence of proficient techniques with the lake, to obtain pure water in your glass.

Do not stress, as by the end of this book, you will have shared in more than 37 years of working experience with data and extracting actionable business knowledge. I will share with you the experience I gained in working with data on an international scale.

You will be offered a processing framework that I use on a regular basis to tame data lakes and the collection of monsters that live in and around those lakes.

I have included examples at the end of each chapter, along with code, that more serious data scientists can use as you progress through the book. Note, however, that it is not required for you to complete the examples in order to understand the concepts in each chapter.

So, welcome to a walk-through of a characteristic data lake project, using practical data science techniques and insights.

The objective of the rest of this introduction is to explain the fundamentals of data science.

Data Science

In 1960, Peter Naur started using the term *data science* as a substitute for *computer science*. He stated that to work with data, you require more than just computer science. I agree with his declaration.

Data science is an interdisciplinary science that incorporates practices and methods with actionable knowledge and insights from data in heterogeneous schemas (structured, semi-structured, or unstructured). It amalgamates the scientific fields of data exploration with thought-provoking research fields such as data engineering, information science, computer science, statistics, artificial intelligence, machine learning, data mining, and predictive analytics.

For my part, as I enthusiastically research the future use of data science, by translating multiple data lakes, I have gained several valuable insights. I will explain these with end-to-end examples and share my insights on data lakes. This book explains vital elements from these sciences that you will use to process your data lake into actionable knowledge. I will guide you through a series of recognized science procedures for data lakes. These core skills are a key set of assets to perfect as you begin your encounters with data science.

Data Analytics

Data analytics is the science of fact-finding analysis of raw data, with the goal of drawing conclusions from the data lake. Data analytics is driven by certified algorithms to statistically define associations between data that produce insights.

The perception of certified algorithms is exceptionally significant when you want to convince other business people of the importance of the data insights you have gleaned.

Note You should not be surprised if you are regularly asked the following: Substantiate it! How do you know it is correct?

The best answer is to point to a certified and recognized algorithm that you have used. Associate the algorithm to your business terminology to achieve success with your projects.

Machine Learning

The business world is buzzing with activities and ideas about machine learning and its application to numerous business environments. Machine learning is the capability of systems to learn without explicit software development. It evolved from the study of pattern recognition and computational learning theory.

The impact is that, with the appropriate processing and skills, you can augment your own data capabilities. Training enables a processing environment to complete several magnitudes of discoveries in the time it takes to have a cup of coffee.

Note Work smarter, not harder! Offload your data science to machines. They are faster and more consistent in processing your data lakes.

This skill is an essential part of achieving major gains in shortening the data-to-knowledge cycle. This book will cover the essential practical ground rules in later chapters.

Data Mining

Data mining is processing data to isolate patterns and establish relationships between data entities within the data lake. For data mining to be successful, there is a small number of critical data-mining theories that you must know about data patterns.

In later chapters, I will expand on how you can mine your data for insights. This will help you to discover new actionable knowledge.

Statistics

Statistics is the study of the collection, analysis, interpretation, presentation, and organization of data. Statistics deals with all aspects of data, including the planning of data collection, in terms of the design of surveys and experiments.

Data science and statistics are closely related. I will show you how to run through series of statistics models covering data collection, population, and samples to enhance your data science deliveries.

This book devotes later chapters to how you amalgamate these into an effective and efficient process.

Algorithms

An algorithm is a self-contained step-by-step set of processes to achieve a specific outcome. Algorithms execute calculations, data processing, or automated reasoning tasks with repeatable outcomes.

Algorithms are the backbone of the data science process. You should assemble a series of methods and procedures that will ease the complexity and processing of your specific data lake.

I will discuss numerous algorithms and good practices for performing practical data science throughout the book.

Data Visualization

Data visualization is your key communication channel with the business. It consists of the creation and study of the visual representation of business insights. Data science's principal deliverable is visualization. You will have to take your highly technical results and transform them into a format that you can show to non-specialists.

The successful transformation of data results to actionable knowledge is a skill set I will cover in detail in later chapters. If you master the visualization skill, you will be most successful in data science.

Storytelling

Data storytelling is the process of translating data analyses into layperson's terms, in order to influence a business decision or action. You can have the finest data science, but without the business story to translate your findings into business-relevant actions, you will not succeed.

I will provide details and practical insights into what to check for to ensure that you have the proper story and actions.

What Next?

I will demonstrate, using the core knowledge of the underlining science, how you can make a competent start to handle the transformation process of your data lake into actionable knowledge. The sole requirement is to understand the data science of your own data lake. Start rapidly to discover what data science reveals about your business. You are the master of your own data lake.

You will have to build familiarity with the data lake and what is flowing into the structure. My advice is to apply the data science on smaller scale activities, for insights from the data lake.

Note Experiment—push the boundaries of your insights.

CHAPTER 1

Data Science Technology Stack

The Data Science Technology Stack covers the data processing requirements in the Rapid Information Factory ecosystem. Throughout the book, I will discuss the stack as the guiding pattern.

In this chapter, I will help you to recognize the basics of data science tools and their influence on modern data lake development. You will discover the techniques for transforming a data vault into a data warehouse bus matrix. I will explain the use of Spark, Mesos, Akka, Cassandra, and Kafka, to tame your data science requirements.

I will guide you in the use of elastic search and MQTT (MQ Telemetry Transport), to enhance your data science solutions. I will help you to recognize the influence of R as a creative visualization solution. I will also introduce the impact and influence on the data science ecosystem of such programming languages as R, Python, and Scala.

Rapid Information Factory Ecosystem

The Rapid Information Factory ecosystem is a convention of techniques I use for my individual processing developments. The processing route of the book will be formulated on this basis, but you are not bound to use it exclusively. The tools I discuss in this chapter are available to you without constraint. The tools can be used in any configuration or permutation that is suitable to your specific ecosystem.

I recommend that you begin to formulate an ecosystem of your own or simply adopt mine. As a prerequisite, you must become accustomed to a set of tools you know well and can deploy proficiently.

A. F. Vermeulen, *Practical Data Science*, https://doi.org/10.1007/978-1-4842-3054-1_1

Note Remember: Your data lake will have its own properties and features, so adopt your tools to those particular characteristics.

Data Science Storage Tools

This data science ecosystem has a series of tools that you use to build your solutions. This environment is undergoing a rapid advancement in capabilities, and new developments are occurring every day.

I will explain the tools I use in my daily work to perform practical data science. Next, I will discuss the following basic data methodologies.

Schema-on-Write and Schema-on-Read

There are two basic methodologies that are supported by the data processing tools. Following is a brief outline of each methodology and its advantages and drawbacks.

Schema-on-Write Ecosystems

A traditional relational database management system (RDBMS) requires a schema before you can load the data. To retrieve data from my structured data schemas, you may have been running standard SQL queries for a number of years.

Benefits include the following:

- In traditional data ecosystems, tools assume schemas and can only work once the schema is described, so there is only one view on the data.
- The approach is extremely valuable in articulating relationships between data points, so there are already relationships configured.
- It is an efficient way to store “dense” data.
- All the data is in the same data store.

On the other hand, schema-on-write isn't the answer to every data science problem. Among the downsides of this approach are that

- Its schemas are typically purpose-built, which makes them hard to change and maintain.
- It generally loses the raw/atomic data as a source for future analysis.
- It requires considerable modeling/implementation effort before being able to work with the data.
- If a specific type of data can't be stored in the schema, you can't effectively process it from the schema.

At present, schema-on-write is a widely adopted methodology to store data.

Schema-on-Read Ecosystems

This alternative data storage methodology does not require a schema before you can load the data. Fundamentally, you store the data with minimum structure. The essential schema is applied during the query phase.

Benefits include the following:

- It provides flexibility to store unstructured, semi-structured, and disorganized data.
- It allows for unlimited flexibility when querying data from the structure.
- Leaf-level data is kept intact and untransformed for reference and use for the future.
- The methodology encourages experimentation and exploration.
- It increases the speed of generating fresh actionable knowledge.
- It reduces the cycle time between data generation to availability of actionable knowledge.

Schema-on-read methodology is expanded on in Chapter 6.

I recommend a hybrid between schema-on-read and schema-on-write ecosystems for effective data science and engineering. I will discuss in detail why this specific ecosystem is the optimal solution when I cover the functional layer's purpose in data science processing.

Data Lake

A data lake is a storage repository for a massive amount of raw data. It stores data in native format, in anticipation of future requirements. You will acquire insights from this book on why this is extremely important for practical data science and engineering solutions. While a schema-on-write data warehouse stores data in predefined databases, tables, and records structures, a data lake uses a less restricted schema-on-read-based architecture to store data. Each data element in the data lake is assigned a distinctive identifier and tagged with a set of comprehensive metadata tags.

A data lake is typically deployed using distributed data object storage, to enable the schema-on-read structure. This means that business analytics and data mining tools access the data without a complex schema. Using a schema-on-read methodology enables you to load your data as is and start to get value from it instantaneously.

I will discuss and provide more details on the reasons for using a schema-on-read storage methodology in Chapters 6–11.

For deployment onto the cloud, it is a cost-effective solution to use Amazon's Simple Storage Service (Amazon S3) to store the base data for the data lake. I will demonstrate the feasibility of using cloud technologies to provision your data science work. It is, however, not necessary to access the cloud to follow the examples in this book, as they can easily be processed using a laptop.

Data Vault

Data vault modeling, designed by Dan Linstedt, is a database modeling method that is intentionally structured to be in control of long-term historical storage of data from multiple operational systems. The data vaulting processes transform the schema-on-read data lake into a schema-on-write data vault. The data vault is designed into the schema-on-read query request and then executed against the data lake.

I have also seen the results stored in a schema-on-write format, to persist the results for future queries. The techniques for both methods are discussed in Chapter 9. At this point, I expect you to understand only the rudimentary structures required to formulate a data vault.

The structure is built from three basic data structures: hubs, inks, and satellites. Let's examine the specific data structures, to clarify why they are compulsory.

Hubs

Hubs contain a list of unique business keys with low propensity to change. They contain a surrogate key for each hub item and metadata classification of the origin of the business key.

The hub is the core backbone of your data vault, and in Chapter 9, I will discuss in more detail how and why you use this structure.

Links

Associations or transactions between business keys are modeled using link tables. These tables are essentially many-to-many join tables, with specific additional metadata.

The link is a singular relationship between hubs to ensure the business relationships are accurately recorded to complete the data model for the real-life business. In Chapter 9, I will explain how and why you would require specific relationships.

Satellites

Hubs and links form the structure of the model but store no chronological characteristics or descriptive characteristics of the data. These characteristics are stored in appropriated tables identified as satellites.

Satellites are the structures that store comprehensive levels of the information on business characteristics and are normally the largest volume of the complete data vault data structure. In Chapter 9, I will explain how and why these structures work so well to model real-life business characteristics.

The appropriate combination of hubs, links, and satellites helps the data scientist to construct and store prerequisite business relationships. This is a highly in-demand skill for a data modeler.

The transformation to this schema-on-write data structure is discussed in detail in Chapter 9, to point out why a particular structure supports the processing methodology. I will explain in that chapter why you require particular hubs, links, and satellites.

Data Warehouse Bus Matrix

The Enterprise Bus Matrix is a data warehouse planning tool and model created by Ralph Kimball and used by numerous people worldwide over the last 40+ years. The bus matrix and architecture builds upon the concept of conformed dimensions that are interlinked by facts.

The data warehouse is a major component of the solution required to transform data into actionable knowledge. This schema-on-write methodology supports business intelligence against the actionable knowledge. In Chapter 10, I provide more details on this data tool and give guidance on its use.

Data Science Processing Tools

Now that I have introduced data storage, the next step involves processing tools to transform your data lakes into data vaults and then into data warehouses. These tools are the workhorses of the data science and engineering ecosystem. Following are the recommended foundations for the data tools I use.

Spark

Apache Spark is an open source cluster computing framework. Originally developed at the AMP Lab of the University of California, Berkeley, the Spark code base was donated to the Apache Software Foundation, which now maintains it as an open source project. This tool is evolving at an incredible rate.

IBM is committing more than 3,500 developers and researchers to work on Spark-related projects and formed a dedicated Spark technology center in San Francisco to pursue Spark-based innovations.

SAP, Tableau, and Talend now support Spark as part of their core software stack. Cloudera, Hortonworks, and MapR distributions support Spark as a native interface.

Spark offers an interface for programming distributed clusters with implicit data parallelism and fault-tolerance. Spark is a technology that is becoming a de-facto standard for numerous enterprise-scale processing applications.

I discovered the following modules using this tool as part of my technology toolkit.

Spark Core

Spark Core is the foundation of the overall development. It provides distributed task dispatching, scheduling, and basic I/O functionalities.

This enables you to offload the comprehensive and complex running environment to the Spark Core. This safeguards that the tasks you submit are accomplished as anticipated. The distributed nature of the Spark ecosystem enables you to use the same processing request on a small Spark cluster, then on a cluster of thousands of nodes, without any code changes. In Chapter 10, I will discuss how you accomplish this.

Spark SQL

Spark SQL is a component on top of the Spark Core that presents a data abstraction called Data Frames. Spark SQL makes accessible a domain-specific language (DSL) to manipulate data frames. This feature of Spark enables ease of transition from your traditional SQL environments into the Spark environment. I have recognized its advantage when you want to enable legacy applications to offload the data from their traditional relational-only data storage to the data lake ecosystem.

Spark Streaming

Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. Spark Streaming has built-in support to consume from Kafka, Flume, Twitter, ZeroMQ, Kinesis, and TCP/IP sockets. The process of streaming is the primary technique for importing data from the data source to the data lake.

Streaming is becoming the leading technique to load from multiple data sources. I have found that there are connectors available for many data sources. There is a major drive to build even more improvements on connectors, and this will improve the ecosystem even further in the future.

In Chapters 7 and 11, I will discuss the use of streaming technology to move data through the processing layers.

MLlib Machine Learning Library

Spark MLlib is a distributed machine learning framework used on top of the Spark Core by means of the distributed memory-based Spark architecture.

In Spark 2.0, a new library, `spark.mk`, was introduced to replace the RDD-based data processing with a DataFrame-based model. It is planned that by the introduction of Spark 3.0, only DataFrame-based models will exist.

Common machine learning and statistical algorithms have been implemented and are shipped with MLlib, which simplifies large-scale machine learning pipelines, including

- Dimensionality reduction techniques, such as singular value decomposition (SVD) and principal component analysis (PCA)
- Summary statistics, correlations, stratified sampling, hypothesis testing, and random data generation
- Collaborative filtering techniques, including alternating least squares (ALS)
- Classification and regression: support vector machines, logistic regression, linear regression, decision trees, and naive Bayes classification
- Cluster analysis methods, including k-means and latent Dirichlet allocation (LDA)
- Optimization algorithms, such as stochastic gradient descent and limited-memory BFGS (L-BFGS)
- Feature extraction and transformation functions

In Chapter 10, I will discuss the use of machine learning proficiency to support the automatic processing through the layers.

GraphX

GraphX is a powerful graph-processing application programming interface (API) for the Apache Spark analytics engine that can draw insights from large data sets. GraphX provides outstanding speed and capacity for running massively parallel and machine-learning algorithms.

The introduction of the graph-processing capability enables the processing of relationships between data entries with ease. In Chapters 9 and 10, I will discuss the use of a graph database to support the interactions of the processing through the layers.