Venkat Ankam

Big Data Analytics

A handy reference guide for data analysts and data scientists to help to obtain value from big data analytics using Spark on Hadoop clusters

Packt

Big Data Analytics

A handy reference guide for data analysts and data scientists to help to obtain value from big data analytics using Spark on Hadoop clusters

Venkat Ankam

BIRMINGHAM - MUMBAI

Big Data Analytics

First published: September 2016

Production reference: 12309016

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78588-469-6

www.packtpub.com

Credits

Author
Venkat Ankam

Reviewers
Sreekanth Jella
De Witte Dieter

Commissioning Editor
Akram Hussain

Acquisition Editors
Ruchita Bhansali
Tushar Gupta

Content Development Editor
Sumeet Sawant

Technical Editor
Pranil Pathare

Copy Editors
Vikrant Phadke
Vibha Shukla

Project Coordinator
Shweta H Birwatkar

Proofreader
Safis Editing

Indexer
Mariammal Chettiyar

Graphics
Kirk D'Penha

Production Coordinator
Arvindkumar Gupta

Cover Work
Arvindkumar Gupta

About the Author

Venkat Ankam has over 18 years of IT experience and over 5 years in big data technologies, working with customers to design and develop scalable big data applications. Having worked with multiple clients globally, he has tremendous experience in big data analytics using Hadoop and Spark.

He is a Cloudera Certified Hadoop Developer and Administrator and also a Databricks Certified Spark Developer. He is the founder and presenter of a few Hadoop and Spark meetup groups globally and loves to share knowledge with the community.

Venkat has delivered hundreds of trainings, presentations, and white papers in the big data sphere. While this is his first attempt at writing a book, many more books are in the pipeline.

Acknowledgement

I would like to thank Databricks for providing me with training in Spark in early 2014 and an opportunity to deepen my knowledge of Spark.

I would also like to thank Tyler Allbritton, principal architect, big data, cloud and analytics solutions at Tectonic, for providing me support in big data analytics projects and extending his support when writing this book.

Then, I would like to thank Mani Chhabra, CEO of Cloudwick, for encouraging me to write this book and providing the support I needed. Thanks to Arun Sirimalla, big data champion at Cloudwick, and Pranabh Kumar, big data architect at InsideView, who provided excellent support and inspiration to start meetups throughout India in 2011 to share knowledge of Hadoop and Spark.

Then I would like to thank Ashrith Mekala, solution architect at Cloudwick, for his technical consulting help.

This book started with a small discussion with Packt Publishing's acquisition editor Ruchita Bansali. I am really thankful to her for inspiring me to write this book. I am thankful to Kajal Thapar, content development editor at Packt Publishing, who then supported the entire journey of this book with great patience to refine it multiple times and get it to the finish line.

I would also like to thank Sumeet Sawant, Content Development Editor and Pranil Pathare, Technical Editor for their support in implementing Spark 2.0 changes.

I dedicate this book to my family and friends. Finally, this book would not have completed without the support from my wife, Srilatha, and my kids, Neha and Param, who cheered and encouraged me throughout the journey of this book.

About the Reviewers

Sreekanth Jella is a senior Hadoop and Spark developer with more than 11 years of IT industry development experience. He is a postgraduate from the University College of Engineering, Osmania University, with computer applications as major. He has worked in the USA, Turkey, and India and with clients such as AT&T, Cricket Communications, and Turk Telecom. Sreekanth has vast development experience with Java/J2EE technologies and web technologies as well. He is tech savvy and passionate about programming. In his words, "*Coding is an art and code is fun* ☺".

De Witte Dieter received his master's degree in civil engineering (applied physics) from Ghent University in 2008. During his master's, he became really interested in designing algorithms to tackle complex problems.

In April 2010, he was recruited as the first bioinformatics PhD student at IBCN-iMinds. Together with his colleagues, he designed high-performance algorithms in the area of DNA sequence analysis using Hadoop and MPI. Apart from developing and designing algorithms, an important part of the job was data mining, for which he mainly used Matlab. Dieter was also involved in teaching activities around Java/Matlab to first-year bachelor of engineering students.

From May 2014 onwards, he has been working as a big data scientist for Archimiddle (Cronos group). He worked on a big data project with Telenet, part of Liberty Global. Working in a Hadoop production environment together with a talented big data team, he considered it really rewarding and it made him confident in using the Cloudera Hadoop stack. Apart from consulting, he also conducted workshops and presentations on Hadoop and machine learning.

In December 2014, Dieter joined iMinds Data Science Lab, where he was responsible for research activities and consultancy with respect to big data analytics. He is currently teaching a course on big data science to master's students in computer science and statistics and doing consultancy on scalable semantic query systems.

I would like to thank iMinds Data Science Lab for all the opportunities and challenges they offer me.

www.PacktPub.com

eBooks, discount offers, and more

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `customercare@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`https://www2.packtpub.com/books/subscription/packtlib`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Table of Contents

Preface

Big Data Analytics aims at providing the fundamentals of Apache Spark and Hadoop, and how they are integrated together with most commonly used tools and techniques in an easy way. All Spark components (Spark Core, Spark SQL, DataFrames, Datasets, Conventional Streaming, Structured Streaming, MLLib, GraphX, and Hadoop core components), HDFS, MapReduce, and Yarn are explored in great depth with implementation examples on Spark + Hadoop clusters.

The Big Data Analytics industry is moving away from MapReduce to Spark. So, the advantages of Spark over MapReduce are explained in great depth to reap the benefits of in-memory speeds. The DataFrames API, the Data Sources API, and the new Dataset API are explained for building Big Data analytical applications. Real-time data analytics using Spark Streaming with Apache Kafka and HBase is covered to help in building streaming applications. New structured streaming concept is explained with an Internet of Things (IOT) use case. Machine learning techniques are covered using MLLib, ML Pipelines and SparkR; Graph Analytics are covered with GraphX and GraphFrames components of Spark.

This book also introduces web based notebooks such as Jupyter, Apache Zeppelin, and data flow tool Apache NiFi to analyze and visualize data, offering Spark as a Service using Livy Server.

What this book covers

Chapter 1, *Big Data Analytics at a 10,000-Foot View*, provides an approach to Big Data analytics from a broader perspective and introduces tools and techniques used on Apache Hadoop and Apache Spark platforms, with some of most common use cases.

Chapter 2, *Getting Started with Apache Hadoop and Apache Spark*, lays the foundation for Hadoop and Spark platforms with an introduction. This chapter also explains how Spark is different from MapReduce and how Spark on the Hadoop platform is beneficial. Then it helps you get started with the installation of clusters and setting up tools needed for analytics.

Chapter 3, *Deep Dive into Apache Spark*, covers deeper concepts of Spark such as Spark Core internals, how to use pair RDDs, the life cycle of a Spark program, how to build Spark applications, how to persist and cache RDDs, and how to use Spark Resource Managers (Standalone, Yarn, and Mesos).

Chapter 4, *Big Data Analytics with Spark SQL, DataFrames, and Datasets*, covers the Data Sources API, the DataFrames API, and the new Dataset API. There is a special focus on why DataFrame API is useful and analytics of DataFrame API with built-in sources (Csv, Json, Parquet, ORC, JDBC, and Hive) and external sources (such as Avro, Xml, and Pandas). Spark-on-HBase connector explains how to analyze HBase data in Spark using DataFrames. It also covers how to use Spark SQL as a distributed SQL engine.

Chapter 5, *Real-Time Analytics with Spark Streaming and Structured Streaming*, provides the meaning of real-time analytics and how Spark Streaming is different from other real-time engines such as Storm, trident, Flink, and Samza. It describes the architecture of Spark Streaming with input sources and output stores. It covers stateless and stateful stream processing and using receiver-based and direct approach with Kafka as a source and HBase as a store. Fault tolerance concepts of Spark streaming is covered when application is failed at driver or executors. Structured Streaming concepts are explained with an Internet of Things (IOT) use case.

Chapter 6, *Notebooks and Dataflows with Spark and Hadoop*, introduces web-based notebooks with tools such as Jupyter, Zeppelin, and Hue. It introduces the Livy REST server for building Spark as a service and for sharing Spark RDDs between multiple users. It also introduces Apache NiFi for building data flows using Spark and Hadoop.

Chapter 7, *Machine Learning with Spark and Hadoop*, aims at teaching more about the machine learning techniques used in data science using Spark and Hadoop. This chapter introduces machine learning algorithms used with Spark. It covers spam detection, implementation, and the method of building machine learning pipelines. It also covers machine learning implementation with H20 and Hivemall.

Chapter 8, *Building Recommendation Systems with Spark and Mahout*, covers collaborative filtering in detail and explains how to build real-time recommendation engines with Spark and Mahout.

Chapter 9, *Graph Analytics with GraphX*, introduces graph processing, how GraphX is different from Giraph, and various graph operations of GraphX such as creating graph, counting, filtering, degrees, triplets, modifying, joining, transforming attributes, Vertex RDD, and EdgeRDD operations. It also covers GraphX algorithms such as triangle counting and connected components with a flight analytics use case. New GraphFrames component based on DataFrames is introduced and explained some concepts such as motif finding.

Chapter 10, *Interactive Analytics with SparkR*, covers the differences between R and SparkR and gets you started with SparkR using shell scripts in local, standalone, and Yarn modes. This chapter also explains how to use SparkR with RStudio, DataFrames, machine learning with SparkR, and Apache Zeppelin.

What you need for this book

Practical exercises in this book are demonstrated on virtual machines (VM) from Cloudera, Hortonworks, MapR, or prebuilt Spark for Hadoop for getting started easily. The same exercises can be run on a bigger cluster as well.

Prerequisites for using virtual machines on your laptop:

- RAM: 8 GB and above
- CPU: At least two virtual CPUs
- The latest VMWare player or Oracle VirtualBox must be installed for Windows or Linux OS
- Latest Oracle VirtualBox, or VMWare Fusion for Mac
- Virtualization enabled in BIOS
- Browser: Chrome 25+, IE 9+, Safari 6+, or Firefox 18+ recommended (HDP Sandbox will not run on IE 10)
- Putty
- WinScP

The Python and Scala programming languages are used in chapters, with more focus on Python. It is assumed that readers have a basic programming background in Java, Scala, Python, SQL, or R, with basic Linux experience. Working experience within Big Data environments on Hadoop platforms would provide a quick jump start for building Spark applications.

Who this book is for

Though this book is primarily aimed at data analysts and data scientists, it would help architects, programmers, and Big Data practitioners.

For a data analyst: This is useful as a reference guide for data analysts to develop analytical applications on top of Spark and Hadoop.

For a data scientist: This is useful as a reference guide for building data products on top of Spark and Hadoop.

For an architect: This book provides a complete ecosystem overview, examples of Big Data analytical applications, and helps you architect Big Data analytical solutions.

For a programmer: This book provides the APIs and techniques used in Scala and Python languages for building applications.

For a Big Data practitioner: This book helps you to understand the new paradigms and new technologies and make the right decisions.

Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Spark's default `OFF_HEAP` (experimental) storage is Tachyon."

Most of the examples are executed in Scala, Python and Mahout shells. Any command-line input is written as follows:

```
[root@myhost ~]# pyspark --master spark://sparkmasterhostname:7077
--total-executor-cores 4
```

A block of Python code executed in PySpark shell is shown as follows:

```
>>> myList = ["big", "data", "analytics", "hadoop" , "spark"]
>>> myRDD = sc.parallelize(myList)
>>> myRDD.getNumPartitions()
```

A block of code written in Python Application is shown as follows:

```
from pyspark import SparkConf, SparkContext
conf = (SparkConf()
        .setMaster("spark://masterhostname:7077")
        .setAppName("My Analytical Application")
        .set("spark.executor.memory", "2g"))
sc = SparkContext(conf = conf)
```

New terms and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "In case of VMWare Player, click on **Open a Virtual Machine**, and point to the directory where you have extracted the VM."

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail `feedback@packtpub.com`, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for this book from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

You can download the code files by following these steps:

1. Log in or register to our website using your e-mail address and password.
2. Hover the mouse pointer on the **SUPPORT** tab at the top.
3. Click on **Code Downloads & Errata**.
4. Enter the name of the book in the **Search** box.
5. Select the book for which you're looking to download the code files.
6. Choose from the drop-down menu where you purchased this book from.
7. Click on **Code Download**.

You can also download the code files by clicking on the **Code Files** button on the book's webpage at the Packt Publishing website. This page can be accessed by entering the book's name in the **Search** box. Please note that you need to be logged in to your Packt account.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of:

- WinRAR / 7-Zip for Windows
- Zipeg / iZip / UnRarX for Mac
- 7-Zip / PeaZip for Linux

The code bundle for the book is also hosted on GitHub at `https://github.com/PacktPublishing/big-data-analytics`. We also have other code bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from `http://www.packtpub.com/sites/default/files/downloads/BigDataAnalyticsWithSparkAndHadoop_ColorImages.pdf`.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to `https://www.packtpub.com/books/content/support` and enter the name of the book in the search field. The required information will appear under the **Errata** section.

Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

Questions

If you have a problem with any aspect of this book, you can contact us at `questions@packtpub.com`, and we will do our best to address the problem.

1
Big Data Analytics at a 10,000-Foot View

The goal of this book is to familiarize you with tools and techniques using Apache Spark, with a focus on Hadoop deployments and tools used on the Hadoop platform. Most production implementations of Spark use Hadoop clusters and users are experiencing many integration challenges with a wide variety of tools used with Spark and Hadoop. This book will address the integration challenges faced with **Hadoop Distributed File System** (**HDFS**) and **Yet Another Resource Negotiator** (**YARN**) and explain the various tools used with Spark and Hadoop. This will also discuss all the Spark components—Spark Core, Spark SQL, DataFrames, Datasets, Spark Streaming, Structured Streaming, MLlib, GraphX, and SparkR and integration with analytics components such as Jupyter, Zeppelin, Hive, HBase, and dataflow tools such as NiFi. A real-time example of a recommendation system using MLlib will help us understand data science techniques.

In this chapter, we will approach Big Data analytics from a broad perspective and try to understand what tools and techniques are used on the Apache Hadoop and Apache Spark platforms.

Big Data analytics is the process of analyzing Big Data to provide past, current, and future statistics and useful insights that can be used to make better business decisions.

Big Data analytics is broadly classified into two major categories, data analytics and data science, which are interconnected disciplines. This chapter will explain the differences between data analytics and data science. Current industry definitions for data analytics and data science vary according to their use cases, but let's try to understand what they accomplish.

Data analytics focuses on the collection and interpretation of data, typically with a focus on past and present statistics. Data science, on the other hand, focuses on the future by performing explorative analytics to provide recommendations based on models identified by past and present data.

Figure 1.1 explains the difference between data analytics and data science with respect to time and value achieved. It also shows typical questions asked and tools and techniques used. Data analytics has mainly two types of analytics, descriptive analytics and diagnostic analytics. Data science has two types of analytics, predictive analytics and prescriptive analytics. The following diagram explains data science and data analytics:

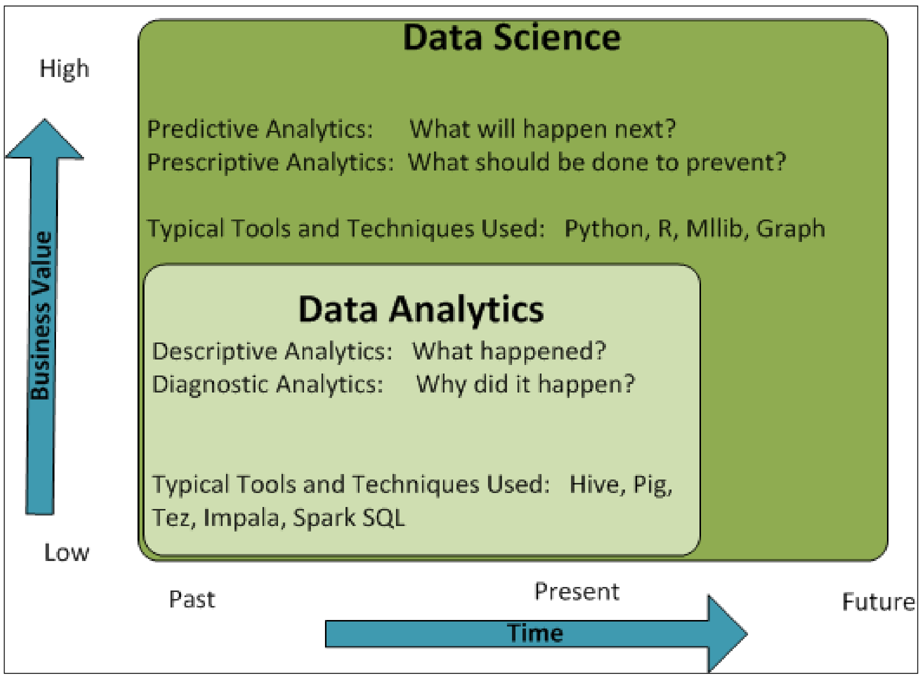

Figure 1.1: Data analytics versus data science

The following table explains the differences with respect to processes, tools, techniques, skill sets, and outputs:

	Data analytics	**Data science**
Perspective	Looking backward	Looking forward
Nature of work	Report and optimize	Explore, discover, investigate, and visualize
Output	Reports and dashboards	Data product

	Data analytics	Data science
Typical tools used	Hive, Impala, Spark SQL, and HBase	MLlib and Mahout
Typical techniques used	ETL and exploratory analytics	Predictive analytics and sentiment analytics
Typical skill set necessary	Data engineering, SQL, and programming	Statistics, machine learning, and programming

This chapter will cover the following topics:

- Big Data analytics and the role of Hadoop and Spark
- Big Data science and the role of Hadoop and Spark
- Tools and techniques
- Real-life use cases

Big Data analytics and the role of Hadoop and Spark

Conventional data analytics uses **Relational Database Management Systems** (**RDBMS**) databases to create data warehouses and data marts for analytics using business intelligence tools. RDBMS databases use the **Schema-on-Write** approach; there are many downsides for this approach.

Traditional data warehouses were designed to **Extract, Transform, and Load** (**ETL**) data in order to answer a set of predefined questions, which are directly related to user requirements. Predefined questions are answered using SQL queries. Once the data is transformed and loaded in a consumable format, it becomes easier for users to access it with a variety of tools and applications to generate reports and dashboards. However, creating data in a consumable format requires several steps, which are listed as follows:

1. Deciding predefined questions.
2. Identifying and collecting data from source systems.
3. Creating ETL pipelines to load the data into the analytic database in a consumable format.

If new questions arise, systems need to identify and add new data sources and create new ETL pipelines. This involves schema changes in databases and the effort of implementation typically ranges from one to six months. This is a big constraint and forces the data analyst to operate in predefined boundaries only.

Transforming data into a consumable format generally results in losing raw/atomic data that might have insights or clues to the answers that we are looking for.

Processing structured and unstructured data is another challenge in traditional data warehousing systems. Storing and processing large binary images or videos effectively is always a challenge.

Big Data analytics does not use relational databases; instead, it uses the **Schema-on-Read** (**SOR**) approach on the Hadoop platform using Hive and HBase typically. There are many advantages of this approach. *Figure 1.2* shows the Schema-on-Write and Schema-on-Read scenarios:

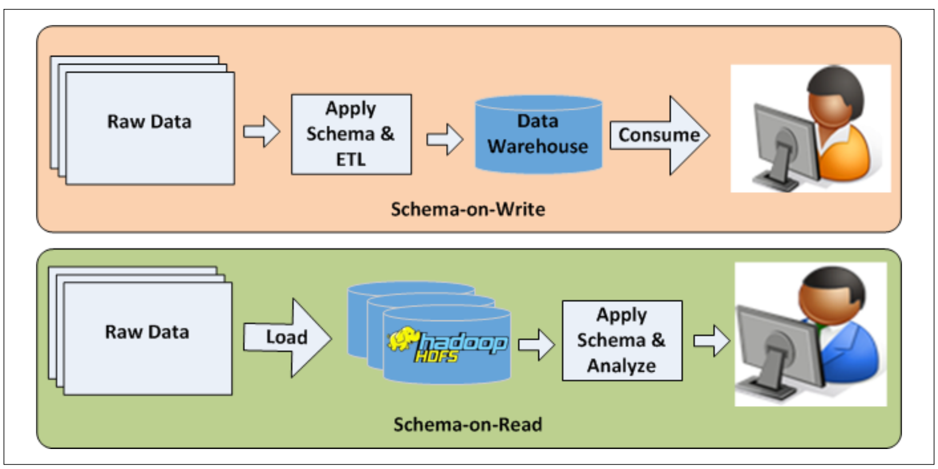

Figure 1.2: Schema-on-Write versus Schema-on-Read

The Schema-on-Read approach introduces flexibility and reusability to systems. The Schema-on-Read paradigm emphasizes storing the data in a raw, unmodified format and applying a schema to the data as needed, typically while it is being read or processed. This approach allows considerably more flexibility in the amount and type of data that can be stored. Multiple schemas can be applied to the same raw data to ask a variety of questions. If new questions need to be answered, just get the new data and store it in a new directory of HDFS and start answering new questions.

This approach also provides massive flexibility over how the data can be consumed with multiple approaches and tools. For example, the same raw data can be analyzed using SQL analytics or complex Python or R scripts in Spark. As we are not storing data in multiple layers, which is needed for ETL, so the storage cost and data movement cost is reduced. Analytics can be done for unstructured and structured data sources along with structured data sources.

A typical Big Data analytics project life cycle

The life cycle of Big Data analytics using Big Data platforms such as Hadoop is similar to traditional data analytics projects. However, a major paradigm shift is using the Schema-on-Read approach for the data analytics.

A Big Data analytics project involves the activities shown in *Figure 1.3*:

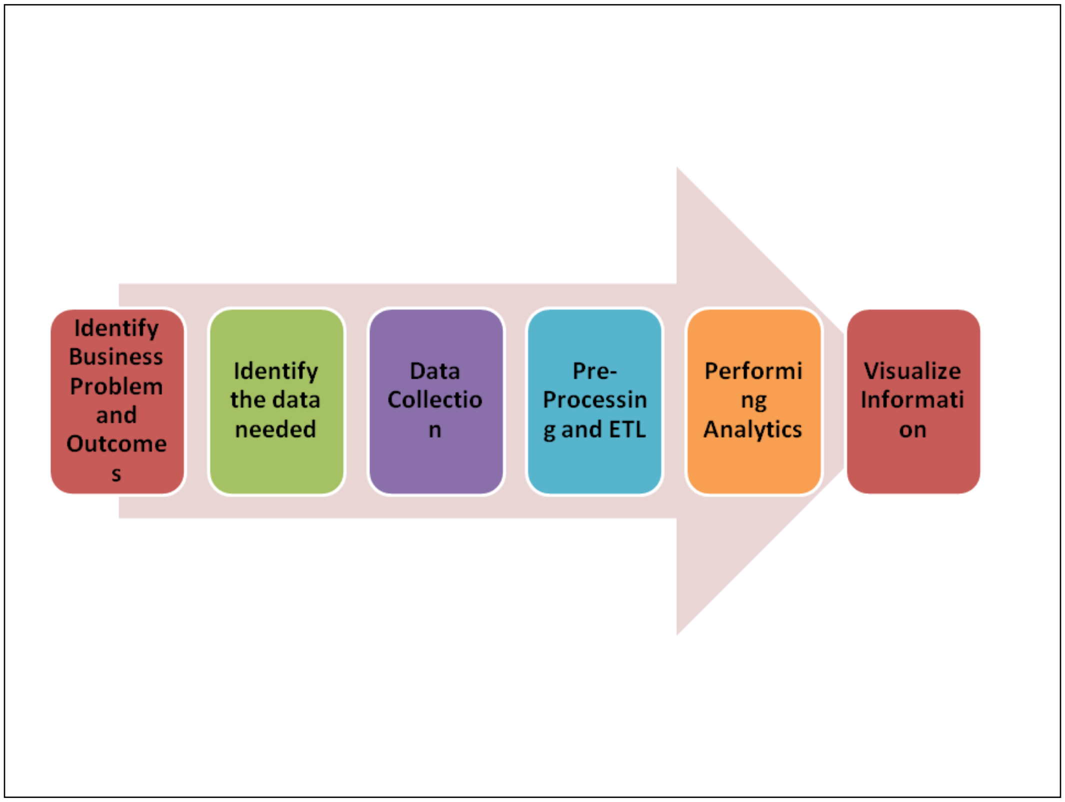

Figure 1.3: The Big Data analytics life cycle

Identifying the problem and outcomes

Identify the business problem and desired outcome of the project clearly so that it scopes in what data is needed and what analytics can be performed. Some examples of business problems are company sales going down, customers visiting the website but not buying products, customers abandoning shopping carts, a sudden rise in support call volume, and so on. Some examples of project outcomes are improving the buying rate by 10%, decreasing shopping cart abandonment by 50%, and reducing support call volume by 50% by the next quarter while keeping customers happy.

Identifying the necessary data

Identify the quality, quantity, format, and sources of data. Data sources can be data warehouses (OLAP), application databases (OLTP), log files from servers, documents from the Internet, and data generated from sensors and network hubs. Identify all the internal and external data source requirements. Also, identify the data anonymization and re-identification requirements of data to remove or mask **personally identifiable information** (**PII**).

Data collection

Collect data from relational databases using the Sqoop tool and stream data using Flume. Consider using Apache Kafka for reliable intermediate storage. Design and collect data considering fault tolerance scenarios.

Preprocessing data and ETL

Data comes in different formats and there can be data quality issues. The preprocessing step converts the data to a needed format or cleanses inconsistent, invalid, or corrupt data. The performing analytics phase will be initiated once the data conforms to the needed format. Apache Hive, Apache Pig, and Spark SQL are great tools for preprocessing massive amounts of data.

This step may not be needed in some projects if the data is already in a clean format or analytics are performed directly on the source data with the Schema-on-Read approach.

Performing analytics

Analytics are performed in order to answer business questions. This requires an understanding of data and relationships between data points. The types of analytics performed are descriptive and diagnostic analytics to present the past and current views on the data. This typically answers questions such as what happened and why it happened. In some cases, predictive analytics is performed to answer questions such as what would happen based on a hypothesis.

Apache Hive, Pig, Impala, Drill, Tez, Apache Spark, and HBase are great tools for data analytics in batch processing mode. Real-time analytics tools such as Impala, Tez, Drill, and Spark SQL can be integrated into traditional business intelligence tools (Tableau, Qlikview, and others) for interactive analytics.

Visualizing data

Data visualization is the presentation of analytics output in a pictorial or graphical format to understand the analysis better and make business decisions based on the data.

Typically, finished data is exported from Hadoop to RDBMS databases using Sqoop for integration into visualization systems or visualization systems are directly integrated into tools such as Tableau, Qlikview, Excel, and so on. Web-based notebooks such as Jupyter, Zeppelin, and Databricks cloud are also used to visualize data by integrating Hadoop and Spark components.

The role of Hadoop and Spark

Hadoop and Spark provide you with great flexibility in Big Data analytics:

- Large-scale data preprocessing; massive datasets can be preprocessed with high performance
- Exploring large and full datasets; the dataset size does not matter
- Accelerating data-driven innovation by providing the Schema-on-Read approach
- A variety of tools and APIs for data exploration

Big Data science and the role of Hadoop and Spark

Data science is all about the following two aspects:

- Extracting deep meaning from the data
- Creating data products

Extracting deep meaning from data means fetching the value using statistical algorithms. A data product is a software system whose core functionality depends on the application of statistical analysis and machine learning to the data. Google AdWords or Facebook's *People You May Know* are a couple of examples of data products.

A fundamental shift from data analytics to data science

A fundamental shift from data analytics to data science is due to the rising need for better predictions and creating better data products.

Let's consider an example use case that explains the difference between data analytics and data science.

Problem: A large telecoms company has multiple call centers that collect caller information and store it in databases and filesystems. The company has already implemented data analytics on the call center data, which provided the following insights:

- Service availability
- The average speed of answering, average hold time, average wait time, and average call time
- The call abandon rate
- The first call resolution rate and cost per call
- Agent occupancy

Now, the telecoms company would like to reduce the customer churn, improve customer experience, improve service quality, and cross-sell and up-sell by understanding the customers in near real-time.