European Funds
Knowledge Education Development

Republic
of Poland

NAWA
POLISH NATIONAL AGENCY
FOR ACADEMIC EXCHANGE

European Union
European Social Fund

Adrian Kapczyński

INTRODUCTION TO BIG DATA

THEORETICAL AND PRACTICAL ASPECTS

Lublin 2021

Introduction to big data

Theoretical and practical aspects

Adrian Kapczyński

Introduction to big data

Theoretical and practical aspects

Adrian Kapczyński

Lublin 2021

Reviewers:

- Prof. Anatoliy Sachenko, PhD.
- Prof. U.S. Jakub Swacha, PhD., Dsc.

Funding: Research, preparation of materials and preparation of the textbook were carried out under the project – grant no. PPI/KAT/2019/1/00015/U/00001 "Cognitive technologies – second-cycle studies in English" and were carried under the KATAMARAN program Polish National Agency for Academic Exchange (NAWA). The program is co-financed by the European Social Fund under the Knowledge Education Development Operational Program, a non-competition project entitled "Supporting the institutional capacity of Polish universities through the creation and implementation of international study programs" implemented under Measure 3.3. Internationalization of Polish higher education, specified in the application for project funding no. POWR.03.03.00-00-PN 16/18.

The project was carried out in cooperation with the Silesian University of Technology (project leader – Poland) and the Kiev National University of Construction and Architecture (project partner – Ukraine).

Project manager: prof. SUT. Aleksandra Kuzior, PhD., Dsc.
Department of Applied Social Sciences
Faculty of Organization and Management
Silesian University of Technology

Typesetting:
Monika Maciąg

Cover designer:
Marcin Szklarczyk

ISBN 978-83-66489-50-9

Publisher:
Wydawnictwo Naukowe TYGIEL sp. z o.o.
ul. Głowackiego 35/341,
20-060 Lublin
www.wydawnictwo-tygiel.pl

To my family and my friends
for their support and encouragement

Table of contents

Introduction

Every company has big data in its future
and every company will eventually be in the data business.

T.H. Davenport

We are living in the complex world which has its (no less) complex digital twin. Looking from the helicopter view, we can see people, machines and connections between them. Emerging megatrends, like social media, mobile or cloud computing are transforming the reality that surrounds us.

If time travel was possible, every jump in time could bring to us a better understanding of human needs and the way they meet them. One of these needs is related to messages we want to pass to future human generations. It brings the questions related to WHAT (message) and HOW (including the way we want to represent that message).

Nowadays, we have advanced means of communication and messages are generated with amazing frequency.

"The Internet Minute" project has shown in its latest update (performed on 10.03.2020), what happens in an Internet Minute in the year of 2020 [1.63]. Among others, we can see the following numbers: 59 million messages sent via WhatsApp, 19 million texts sent, 4.7 million videos viewed on YouTube, 1.6 million swipes on Tinder and 305 smart speakers (like Amazon Echo or Google Home) are shipped to customers who have ordered them.

We are generating a lot of data every minute and the data must be transmitted and stored (sometimes for a shorter time and sometimes for a longer time). We demand more and more network bandwidth and more and more disk space (on our local devices or in the cloud). The digital traces we create (for example during electronic shopping) can be the basis of the analysis, for example for the purposes of minimizing the probability of abandoning the shopping cart. Well, we are very close to showing the main area of the interest of this book: the big data.

The main purpose of this handbook is to provide basic basic knowledge about big data issues, both from a theoretical and practical point of view. The prepared study is intended for a wide range of readers, however it is assumed, that if the reader requires additional knowledge, then he (or she) explores it by studying references sources and other sources that may become useful. In short, we utilize the scientific-based learning approach.

This book is divided into the following main parts: introduction, main part divided into chapters, closing remarks, references and appendix.

Chapter one is devoted to the theory of big data, while the next two chapters are related with practical aspects of big data (Chapter 2 and Chapter 3). In closing remarks we show areas of interest which were not covered in the book, but they are worth to be considered to explore after finishing reading this book.

We assume that examples are not just another way of explanation, but we treat them as the key way to understand the issues presented in this book. We encourage our readers to explore the literature references as well as move further, keeping in mind, that the topic of big data is developing dynamically. In simple words, it is worth not to stick with state-of-the art provided in the chapters, but to complete them with new progress in research work.

This book has been written for the teaching purposes related to a course of study entitled: "Cognitive Technologies". This project is funded by Polish National Agency for Academic Exchange. Hereby, I would like to express my appreciation to prof. Aleksandra Kuzior for offering me the possibility to become a member of the project team.

ACKNOWLEDGEMENT

I would like to express my gratitude to my colleagues from the MMI Research Team at Silesian University of Technology, especially I would like to thank Mr. Piotr Halama for his technical support that was invaluable from my point of view. I would like to thank the reviewers for their valuable remarks.

SHARING YOUR OPINION

This is the first edition of this book. It would be more than welcome if you decide to share your insights, ideas or spotted erros, thereby helping me in preparation better quality of the next edition of this book.

Contact details: Adrian Kapczynski, adriank@polsl.pl.

ADDITIONAL MATERIALS

This book is equipped with additional electronic materials, which are published on Github platform. In the appendix located in the closing part of the book, we can find more details about it.

Please follow presented below URL to find more:

https://github.com/adriank-ps/bigdatabook-v1.

Chapter 1. Big data: theoretical fundamentals and real-life examples

In God we trust. All others must bring data.

W. Edwards Deming

1.1. INTRODUCTION

Let us begin our journey through the digital world of big data with a simple question that sounds: "What is big data?". Big data is a buzzword, that combines *large data sets* and *ecosystem* that is required for the storage and processing of those data sets. It is recommended to start with the following sources [1.7], [1.26], [1.36] and [1.57].

Naturally, we can start to think of the big data through real-world application cases.

In subchapter (1.7) we will discuss the real-life applications in more detail, but now Let us mention just two examples from China [1.52]. First is China's National Credit Scoring System built on National Personal Credit Database, which consists of all Chinese banking records. A second example is related with internet healthcare solutions, i.e., ChunYu Doctors App. The solution analyzes symptoms, medication history and other information supplied by patients (in unstructured form), the system assigns appropriate subjects and is able to recommend the best on-line doctor. According to literature, the latter big data platform is supporting more than 500 000 doctors and more than 200 000 000 patients.

Next, let us think about big data from an organizational level.

In [1.36] we can find a valuable description of a traditional data warehouse (which is a system used for reporting and data analysis), as a base for characterizing the *big data warehouse*.

The data processing life cycle starts with the definition of data sources (based on identified business needs), then comes the phase of data integration. Going further, it is important to perform data quality analysis (to be sure that data is correct and complete).The next phase is related to building data models and finally, we go into the phase of development of reporting and analysis applications.

Looking at the comparative view of *big data warehouse* and classical *enterprise data warehouse*, we shall consider business expectations (e.g., focusing on exploratory analysis), design methodology (e.g., highly agile), data architecture (e.g., with ability to scale) and other (e.g., high distribution of data processing programs).

Finally, in case of big data solutions, we will require technologies which will be able to: deal with a variety of data types, deal with data *at rest* and data *at motion*, manage distributed data, integrate any source of data which structure is not known and other, like provisioning high availability.

There are other thought-provoking issues related to big data. For example, in proceedings of The First International DASFAA Workshop on Big Data Management and Analytics (BDMA 2013) [1.7], we can find an interesting discussion about challenges, issues and opportunities related to big data *mining*. At the beginning of the article a few important things are emphasized:

1. Due to the technology revolution, including access to the Internet and the introduction of a great range of digital devices, such as remote sensors, a tremendous amount of data is generated.
2. We have a great variety of types of data: text, images, sounds or anything that is a combination of basic types.

3. Data is a basic component of data streams, which usually implies a real-time manner.

Thinking about the list presented above, we can see three main keywords: volume, variety and velocity. Authors starting from these three main characteristics of big data are indicating *big data mining* as the main area of further investigation. Big data mining begins with data selection and after this comes the need for development of new ways of data filtering, data cleaning, data reduction and so on.

The key challenges of big data mining are connected with:

a) *variety and heterogeneity*, including the question about how to construct a complex, multi-model system,
b) *scalability*, including the question about how to navigate through search space,
c) *trust*, including the question about the possibility of verification of given data source, as they are of different origins and some of them are not well known,
d) *interactiveness*, including the question about how to deal with user interaction (feedback and guidance) for example allowing users to pre-evaluate mining results.

It is worth paying attention to the challenge entitled "Garbage Mining", where big data mining could be applied for data cleaning purposes by mining the garbage and recycling it.

It is important to analyze recent trends of big data, especially in the field of big data platforms and big data applications. For example, it is worth analyzing the case of Hadoop platform which is an open-source implementation of MapReduce, which was based on the paradigm of running big data solutions without the requirement of using expensive high-performance computing platforms, and use commodity servers instead [1.57].

In this chapter, we introduce the formal definition of big data, as well as we briefly describe big data characteristics. We will bring some examples of big data applications (subchapter 1.5). Later, we will discuss big data security and privacy issues (subchapter 1.4). Closely connected with the topic of big data privacy is the question related to moral principles of big data, which we will cover in subchapter 1.5. Big data is an interesting topic from a technical and scientific perspective, so it is reasonable to provide some big data challenges (subchapter 1.6). The subchapter 1.7 is devoted to real-life examples of big data solutions. It is worth to be noticed that it could be very beneficial if included references, which will be individually deepened by the reader of this book. It consists of several dozen items, which broaden the issues signaled within the chapter. In the final section of the chapter, a list of exercises is provided. That list can be useful for both purposes: to indicate the most important topics covered in this chapter and to serve as a teaching to be used as a teaching aid for classes in a subject related to big data.

1.2. BIG DATA DEFINITION

Big data can intuitively be defined as something that has to do with data of large size. For example, we can imagine a phone book with the phone numbers of all residents of the country (e.g., Poland). Sticking to the example which is related to telephony, it is worth considering what phones are used for (in particular we will consider mobile phones). Naturally, mobile phones are used to make phone calls, which on the caller’s side are outgoing calls, and on the receiving side they are incoming calls. It is not the end of the list of possibilities, for example, we can think of short text messages, messages containing multimedia, and so on.

If we take a look from the user's side, we will see single sets of data, e.g., a list of outgoing calls, or a set of received text or multimedia messages. In the era of new possibilities of mobile devices (which can be spotted by analysis of the size of data backups that we store in the phone's memory), we can conclude that there is usually a lot of data and that they have its representation as files which are of different sizes. A shorter text document containing a shopping list will have a smaller size, and a full-length multimedia movie downloaded from Video-On-Demand service will have a larger size.

From the perspective of a single user of a mobile device, we can move to the next level and consider how the dataset looks from the perspective of a telecommunications operator. What a single subscriber sees (as part of his detailed bill for a given month) is the list of all charges for his calls (as well as for other services) is from the operator's point of view, only a subset of a larger dataset. Moreover, in our considerations, it is impossible to ignore the issue of time of storage of telecommunications data, and finally, we may think of other specific questions, for example: how data is stored or how data is made available to institutions requesting access to it. Of course, this example can be further developed, but we will stop here and retain the first, informally worked out elements:

1. We have data that has big size.
2. We have data that has a small size, but can be in large amounts.
3. We have data that can be of various types.
4. There are important issues connected with data, like the security of its storage.

This introductory example shows how starting with things that are commonly known helps to understand the new definition. Moreover, it shows how important it is to use examples that help us develop our *own definition* that we will remember for a longer period of time.

Let us now move on to the definition of *big data*, to which this book is devoted. However, before we consider the definitions proposed by scientists in academic publications, it is worth starting with the information available in the free encyclopedia. This will allow obtaining a relatively complete first look at a set of aspects which are related to *big data*, as the term that is in the area of our interest.

> ***Big data*** *is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software* [1.62].

The article in the encyclopedia devoted to explaining what *big data* is, in addition to the definition itself, contains the characteristics of big data, as well as presents architecture, technologies, and applications related to big data. Presented case studies and research activities are worth reading as well. By reading the text, we can learn that big data is connected with data capturing, data storage, data analysis, and other actions, such as visualization. Next, in addition to issues related to how to collect, process, and present data, ethical and legal aspects should also be taken into account. Initially, big data was explained through the prism of three main characteristics: volume, velocity, and variety. Over time, more characteristics, such as veracity or value, have emerged.

Large data sets are inseparably associated with technical solutions that allow their storage and processing. Among the milestones in the development of big data systems, one can notice such solutions as DBC 1012 (Teradata Corp.), HPCC Systems (Seisint

Inc.), and Hadoop (Apache). Noteworthy is the application section, which refers to the sphere of government administration, healthcare, insurance, media, and IT.

We will discuss the applications in the further subsection of this chapter.

Now, let us take a look at definitions of big data, which were brought by a literature review [1.12], [1.20], [1.28], [1.31] and [1.38].

> *Every day, we create 2.5 quintillion bytes of data – so much that 90% of the data in the world today has been created in the last two years alone. This data comes from everywhere: sensors used to gather climate information, posts to social media sites, [...], and cell phone GPS signals to name a few. This data is big data* [1.28].

Big data is related to the ability to store, process and access all the data that is useful from an organization's perspective. It inspires to formulate three main questions:

e) Can we acquire and store the data?
f) Can we prepare and analyze the data?
g) Can we search, retrieve, visualize the data?

If the answer to the above questions is "Yes", then we are on the best way to understand what *big data means*.

In [1.35] we can come across a systematized list of definitions of big data.

It will be recommended to read carefully those definitions and think about what they have in common and how they differ from each other.

Here is the list of chosen definitions (year published; author(s) last name(s); definition quote):

1. 2011; Russom; "Big data involves the data storage, management, analysis, and visualization of very large and complex datasets".
2. 2011; White; "Big data involves more than simply the ability to handle large volumes of data; instead, it represents a wide range of new analytical technologies and business possibilities. These new systems handle a wide variety of data, from sensor data to Web and social media data, improved analytical capabilities, operational business intelligence that improves business agility by enabling automated real-time actions and intraday decision making, faster hardware and cloud computing including on-demand software-as-a-service. Supporting big data involves combining these technologies to enable new solutions that can bring significant benefits to the business".
3. 2012; Beyer & Laney; "Big data: high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery, and process optimization".
4. 2012; Boyd & Crawford; "Big data: a cultural, technological, and scholarly phenomenon that rests on the interplay of (1) Technology: maximizing computation power and algorithmic accuracy to gather, analyze, link, and compare large datasets. (2) Analysis: drawing on large datasets to identify patterns in order to make economic, social, technical, and legal claims. (3) Mythology: the widespread belief that large datasets offer a higher form of intelligence and knowledge that can generate insights that were previously impossible, with the aura of truth, objectivity, and accuracy".

5. 2014; Davis; "Big data consists of expansive collections of data (large volumes) that are updated quickly and frequently (high velocity) and that exhibit a huge range of different formats and content (wide variety)".
6. 2016; Akter (et al.); "Big data consists of expansive collections of data (large volumes) that are updated quickly and frequently (high velocity) and that exhibit a huge range of different formats and content (wide variety)".

Scientists not only have provided definitions of big data, but also have defined attributes of *big data,* which we can notice in the majority of definitions listed above. First, let us introduce a full list of attributes of big data, that we can come across during literature review [1.35].

Volume is the attribute related to the size of the dataset, defined by a large number of variables (represented by columns) and a large number of observations.

Velocity is the attribute related with the speed of data collection and analysis.

Variety is the attribute related to the plurality of data sources, which can be structured or unstructured.

Veracity is the attribute related to the authenticity of the data and its protection from unauthorized access and modification.

Value is the attribute related to benefits obtained by big data analysis.

In literature, there can be found two more attributes: **Variability** (dynamic opportunities through data interpretation) and **Visualization** (data representation in a meaningful way).

The list of proposed configurations of big data attributes consists of:

- 3V (Volume, Velocity, Variety) – most popular;
- 4V (Volume, Velocity, Variety, Veracity);
- 5V (Volume, Velocity, Variety, Veracity, Value);
- 7V (Volume, Velocity, Variety, Veracity, Value, Variability, Visualization).

Analysis of listed above definitions of the term *big data,* can lead us to a proposal of a formal model of big data, which will be in the form of a quartet [1.38]:

1. Set of volume types.
2. Set of types of data sources.
3. Set of big data analysis techniques.
4. Set of big data processing technologies.

A good illustration of the definition of big data can be found in [1.20], which is based on an example of Customer Relationship Management (CRM) use case. In traditional (non-big data) approach, the main area of interest (in case of CRM systems) is related with such challenges as segmentation of customers or target-group marketing actions (based on age, gender, education, etc.), while in big data approach we can move to the customer-oriented directive, offering cross-selling or up-selling with real-time analysis of the given customer’s behavior, thus make it possible to focus on individual customers’ needs.

In [1.12] we can find a definition of the big *data value chain.* Value chain, as defined by Porter, is related to a set of high-level activities that are performed by the organization to deliver a product (or a service) to the market. Within a value chain, there is a set of subsystems and each subsystem has its inputs, process and outputs. In the context of big data, *big data value chain* consists of the following activities:

a) Data acquisition (data gathering, data filtering and data cleaning),
b) Data analysis (data exploring, data transforming, data modeling),
c) Data curation (data quality enhancements),
d) Data storage (data storage in-memory, NoSQL databases, etc.),
e) Data usage (data analysis and its integration with business activities).

Operating at a higher level of abstraction, we can think of big data ecosystem, which consists of [1.12]:

a) big data suppliers (creating, collecting, transforming data),
b) big data technology providers (providing tools, platforms, etc.),
c) big data end-users (user or organization taking advantage of big data),
d) big data marketplace (hosting data and offering them to customers),
e) big data startup and entrepreneurs (providing new data-driven services or products),
f) big data Researchers (providing new algorithms),
g) Regulators and Standardization bodies (defining law regulations and standards),
h) Investors (providing means to develop the ecosystem).

In [1.31] we can find a comprehensive description of big data architecture.

It is good to start with big data concerns which were divided into five groups:

1. Data security and privacy:

a) Legitimacy of data acquisition,
b) Security of data storage,
c) Security of data transmission,
d) Data security (and privacy) related law regulations,
e) Data security (and privacy) related standards.

2. Infrastructure:

a) Distributed data architecture,
b) Database technology,
c) Cloud services,
d) Data integration of heterogeneous sources,
e) Scalability.

3. Data quality:

a) Data source credibility,
b) Data relevance,
c) Data availability,
d) Data integrity,
e) Data accuracy.

4. Big data analysis:

a) Data analysis logic,
b) Data analysis process,
c) Data analysis algorithm,
d) Data-driven system design.

5. Big data application evaluation:

a) Evaluation method and of big data application,
b) Evaluation index of big data application.

Furthermore, we can find detailed reference models, including:

a) big data performance reference model,
b) business reference model,
c) application reference model,
d) data reference model,
e) infrastructure reference model,
f) architecture security reference model.

Visualization of big data performance reference models was shown in Figure 1.1, while visualization of the data reference model was shown in Figure 1.2.

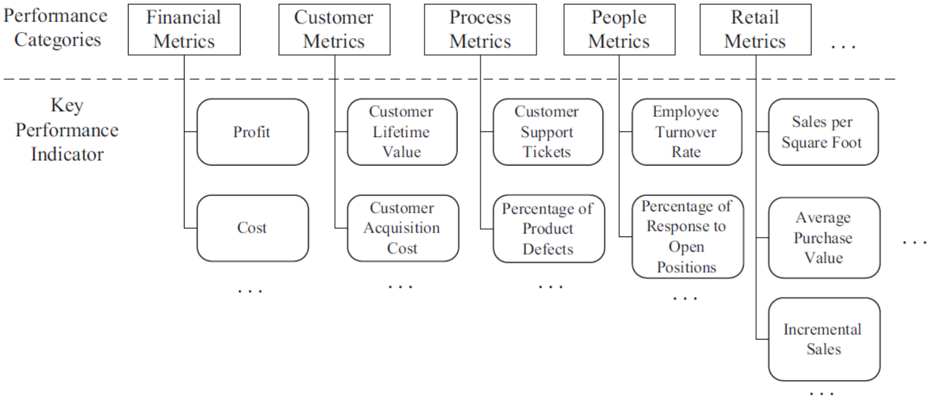


Figure 1.1. Big data architecture: performance reference model. *Source: [1.31]*

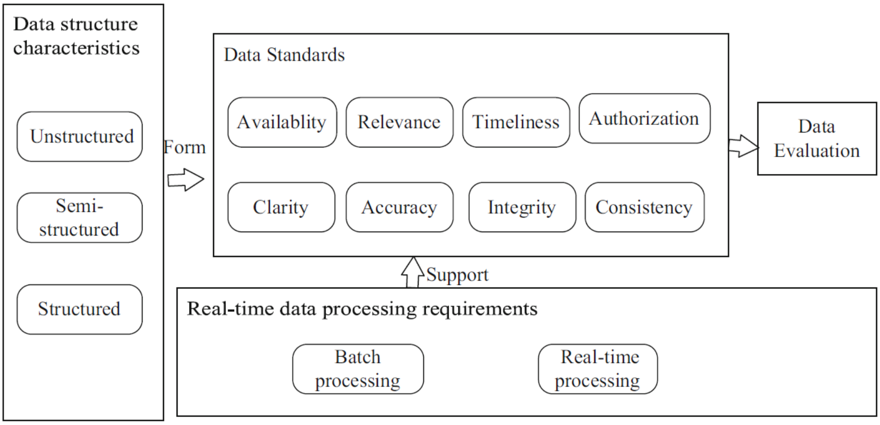


Figure 1.2. Big data architecture: data reference model. *Source: [1.31]*

The use of presented reference models can help solve top-level design problems of big data solutions, as well as it can be used during the evaluation of the implementation of a given big data solution. Of course, more low-level analyses of big data solutions can bring additional value and will be complementary, to top-level analyses.

1.3. BIG DATA APPLICATIONS

As has been mentioned before (see p. 1.2), among the most popular applications of big data are those which are related to (among others): government sector, media, education, and IT.

Naturally, that list of general areas of applications can be deepened by providing a detailed description of big data usage.

According to the literature, big data applications are on different levels of advancement and at various stages of maturity. Let us take a closer look at selected examples.

We can find a lot of documented applications that are connected with big data. Let us browse through the chosen ones.

A. Rambousek (et al.) from Masaryk University, has undertaken the research project whose goal was to investigate the origins of family names from Britain and Ireland [1.47]. The project provided tasks related with the development of software tools (for interaction with big data) and the tasks related with data, including preprocessing steps. The original database contained 188 043 185 records and each record contained inter alia information about the type of event (birth, marriage, death, christening), personal data (first name, last name), date and location. The cleaning process covered deletion of obvious mistakes, standardization of name of the counties, deletion of duplicate records. For sure, we can imagine a similar project, for example, related to data about student relationships with the university.

A specific, however well-documented application of big data, can be found in [1.49]. Authors have decided to unleash the potential of big data in mental healthcare. The paper discusses the main possibilities and limitations of the technique used to improve psychiatric care through big data. It is worth noting that this project has two fundamental pillars: big data and AI (Artificial Intelligence).

As we could see based on the previous example, big data can be combined with AI. We can meet quite often the combination of big data and AI and also the combination of big data and Internet of Things (IoT).

For example, in [1.3] we can read about the application of big data combined with IoT. It is almost a 40-page long document, surely worth reading in full, however, we will focus on just two aspects: a) big data and its sources, and b) big data in IoT Application areas. If it is going about big data and its sources, the following sources of data shall be taken into consideration: RFID (Radio-Frequency Identification), Wireless Sensor Networks, Machine-To-Machine Communications and Cloud computing. Among big data in IoT application areas, we can list the following application areas: healthcare systems (connecting medical equipment, objects and humans), food supply chains, smart power systems or firefighting.

An extremely relevant and interesting source of big data application for traffic monitoring and management, can be found in [1.55]. In this thesis, we can become familiar with the results of three main projects:

- WHAT (Automatic Accounting of Modern Web Services);
- AWESoME (Big Data for Automatic Web Service Management in SDN);
- PAIN (A Passive Web Performance Indicator for ISPs).

It is important to take a closer look at the methodology which was used. This methodology covers the scenario, definitions, architecture, datasets and evaluation.

Further applications of big data include:

- Use of big data for high-energy physics [1.11];
- Use of big data in decision-making (customer intelligence, supply chain management, quality management and risk management) [1.17];
- Use of big data in management the life cycle replacement program for telecommunication power systems' equipment [1.34];
- Use of big data in capital market use cases [1.44];
- Use of big data in text mining of community question answering [1.46].

1.4. BIG DATA SECURITY AND PRIVACY ISSUES

Well, no doubt about it, nowadays, using electronic devices in our everyday life, we are generating a lot of digital traces.

At the beginning of this chapter, we have considered an example related to mobile devices. Let us continue in that field of interest and for a moment let us think of a list of most frequently used mobile applications. Without any scientific investigation, on such a list we can probably find an Internet browser, social media application, electronic mail client, instant messenger or maybe more basic (single-function) applications, like flashlight or timer.

Regardless of which mobile application is used most frequently, the key aspect is related with data we generate while we are using a given application.

Let us consider social media applications. We are not only reading the content, but also we are interacting with it. Even more, we can be active users, enriching the content on the social media platform by posting, commenting or sharing. Last, please consider the use of GPS while you are using your mobile phone, which frequently has a non-stop connection with the Internet. So, we shall consider not only the data stored on mobile devices' memory (e.g., event logs), but also all the data that is transferred (sending and receiving) via a network interface (e.g., application usage reports). Some of that data we may want not to share with anyone due to professional reasons, e.g., it is company confidential data and some of the data may have a specific character (e.g., be personal data or so-called personally identifiable information), which confidentiality shall be protected by law (e.g., due to GDPR law regulations).

And now let us imagine the bigger picture.

First, let us assume that the mentioned mobile device is one of many mobile devices acting as hardware that enables human-machine interaction and network communication.

Second, let us assume that social media applications are just software allowing interaction with multi-user internet platforms. It formulates the digital ecosystem with users whose interactions can be analyzed and based on that, it allows performing some actions (e.g., instant recommendations or decisions about advertising campaigns).

In fact, in the provided example, we are dealing with a big data system and one of the interesting aspects of big data, namely the one which is related to security and privacy.

Let us focus on considerations related to big data security and privacy, formulated after analyzing selected literature items.

Big data offers its potential, but also big data imply risks [1.22]. Big data security shall be analyzed from different levels: from management (high-level) to storage (low-level) [1.56]. It is extremely difficult to address three main security attributes, i.e., confidentiality, integrity and availability to big data, due to the pace of data growth as well as because of a wide range of data processing techniques. Big data storage and processing, due to volume of the data implies increased surface of security attack, as well as make the data leakage the challenge of utmost importance [1.4]. One of the interesting topics regarding big data and security is related to using the big data potential in the malware detection process, mainly in anomaly-based detection of unknown security attacks.

Big data solutions provide security mechanisms which are related to authentication, authorization, auditing and encryption of data at motion as well as at rest. For more information about the above-mentioned mechanisms, please reference [1.34]. It is significant not to forget about providing adequate security mechanisms against data interception attacks. If we assume that our big data solution has a network-based access, it will use such communications protocols as TCP/IP. If yes, then in the scope of our area of interest we should include data communication equipment, like routers of switches. For security purposes, it is crucial to design and evaluate a policy-based security routing and switching [1.23].

In [1.24] we can find the multi-layered security model, which provides a systematic view of security countermeasures implemented to provide data integrity, data authenticity and data confidentiality.

Privacy can be confused with security, however, it is important to understand the difference. Privacy is not about confidentiality, integrity or availability, but it is about the appropriate use of an individual's information. Privacy-preserving methods are based on alteration of the source data: data twisting, data remaking or data encryption. Data anonymization works like sanitizer: we try to make sensitive data protected, by converting the clear data into an unreadable and also irreversible form. Among others, scientists propose the use of 3DES encryption algorithms [1.48]. Particular privacy-preserving techniques are classified into: input data privacy and output data privacy. Please consider [1.50] as the source of details about privacy techniques (e.g., k-anonymity), models (e.g., differential privacy) and frameworks (e.g., EEXCESS).

We have to keep in mind that there exist law regulations related to personal data that shall be obeyed, and among them, one of the most important is GDPR (General Data Protection Regulation). According to the regulation, we have to implement personal data processing rules, e.g., to fulfill the duty to inform. An interesting discussion about GDPR as a (big) genetic data enabler was presented in [1.42].

In [1.30] we can find an interesting case study related to consumer privacy, particularly about US privacy protection acts (Consumer Privacy Bill of Rights, Federal Trade Commission Privacy Report and California Online Privacy Protection Act).

Finally, we can present some interesting challenges related to big data security and big data privacy.

In the area of big data security we can enumerate the following big data security challenges [1.15]:

1. Infrastructure security:

a) Secure computations,
b) Secure storage of data and transaction logs,
c) Input validation.

2. Access control and policy:

a) Granular access control and data access policies,
b) Cryptographically enforced access control.

3. Data management:

a) Real-time security monitoring,
b) Auditing,
c) Data provenance.

4. Privacy:

a) Scalable privacy-preserving data mining.

There is a list of open privacy challenges related to big data [1.27]:

a) Scalable privacy-preserving data analytics,
b) Risk in use of third-party tools,
c) Lack of measures needed to ensure data security and privacy.

1.5. BIG DATA AND ETHICS

Let us imagine that we are running a store and we have introduced loyalty cards.

We are provisioning the loyalty cards to customers after successful fulfillment of the registration form, which consists of fields requesting customer's personal data, e.g., first name, last name, address, etc. Each loyalty card has its unique identifier, frequently represented as two-dimensional barcode.

At first glance, customers may think of such a loyalty card as a non-human related identifier (e.g., it is just a barcode required to obtain benefits) or as a piece of plastic (or its virtual equivalent) which can be used by anyone whom I decide to let it use.

So, some of the customers may think of it not as something that can be useful in characterizing or later to assign them to a certain group of customers (so-called customer profiling).

Without further elaboration, we can pose the question about moral principles, which can be brought here: the right and wrong about what we (as managers of the shop) decide to introduce to our shop.

Let us consider some issues related to big data and ethics based on the performed literature review of the selected literature items.

We are living in the digital era in which ethics not only applies to humans but also to machines [1.42]. One of the examples illustrating that digital ethics is connected with self-driving cars and decisions about pedestrians' lives (please check "MIT Moral Machine").

The technological revolution enabled the formulation of global digital platforms. Data sources and algorithms (including those based on artificial intelligence) are used to (among others): optimize processes, predict patterns or support decision-making processes. One of the examples of such a global digital platform is Twitter. And some questions arise, including the following [1.29]: "*Does it matter if the owners of the sentiments being analyzed in sentiment analysis on Twitter do not know they are being analyzed?*"

According to [1.54] there are four principles that can be used to specify the ethical meaning of privacy. Let us describe them briefly:

a) nonmaleficence is related to harm following the use of personal data,
b) justice is related to the distribution of goods, opportunities among individuals or groups,
c) autonomy is related to the decision-making capacities of individuals (or groups),
d) trust is the relation between the data sources and parties which are using the data.

There are eight major groups of ethical values that are connected with big data and its applications [1.10]: a) contextual integrity, b) controlling identity, c) copyright, d) informational self-determination, e) non-discrimination, f) privacy protection, g) solidarity and h) transparency.

1.6. BIG DATA CHALLENGES

In the field of big data, we can find many interesting challenges. In this subsection, we will focus our attention on interesting (in the author's opinion) cases listed below.

Big data challenges are mainly connected with designing and implementation of big data solutions in such a way that it will enable companies to obtain benefits from it (e.g., creating new services (products) or improving existing ones [1.8].

The key challenges related with big data are caused by its characteristics, which means we have to handle: a large amount of data, high dimensional and high dynamical data [1.25]. From a practical point of view, among big data challenges, we shall list the following: policies and procedures (needed for being compliant with legal requirements), appropriate access to data, dealing with an organization’s legacy systems and lack of technical knowledge about new technologies connected with given big data solutions.

One of the big data challenges is related with Network Traffic Monitoring and Analysis, abbreviated NTMA [1.14]. Network traffic monitoring and analysis is a very complicated task and its characteristics, i.e., volume, velocity, veracity and variety are making it a perfect example of big data challenge. There are four main categories of network traffic monitoring and analysis applications: a) traffic prediction, b) traffic classification, c) fault management and d) network security. Regardless of the area of application, the key aspect is related with data management. Data management includes the following steps: *data collection* by use of packet-based methods or data-flow methods; *data ingestion* which is about data delivery to big data system, data storage (with an important role of horizontal scalability which relies on increasing capacity by utilization of multiple servers) and finally, *data pre-processing* (data transformation, feature engineering).

Another big data challenge was documented in [1.61]. Authors attempted to use big data to analyze the mechanisms of the sharing economy which is the phenomenon of sharing resources among users (e.g.car sharing).

Furthermore, we could encounter [1.2] an interesting challenge related to the cybersecurity of big data systems.

We can distinguish different areas interesting from the security management point of view:

1. Data repository security management (with important role of access control mechanisms).
2. *Data storage security management* (with important role of encryption of storage media).
3. *Distributed file systems security management* (with important role of authentication, authorization and encryption).
4. *Data streams security management* (with important role of access control and encryption of flow of data from data generator to data receiver or data consumer).
5. *Big data computing security management* (with an important role of identity and management mechanisms and data-transmission encryption protocols).

In [1.59] we can find information about big data challenges related to astronomy, precisely with Chinese science project FAST (Five-hundred-meter Aperture Spherical radio Telescope) aimed at pulsar search. The big data system was characterized by: single source, data rate at 6 GB per second and data volume at 20 PB per year.

The challenges related to big data used for Internet of Things consist of [1.13]:

1. Challenges related with data acquisition:

a) Difficulties caused by different data representation,
b) Difficulties related with efficient data transmission,

2. Challenges related with data preprocessing and data storage,

a) Difficulties related with data integration,
b) Difficulties related with reduction of temporal and spatial redundancy,
c) Difficulties related with data cleaning,
d) Difficulties related with data compression.
e) Difficulties related with reliability, persistency, scalability and efficiency of data storage.

3. Challenges related to data analytics.

a) Difficulties caused by data correlation,
b) Difficulties caused by lack of efficient data mining approaches,
c) Difficulties related with security,
d) Difficulties related with privacy.

The paper [1.37] was devoted to discussion about big data challenges in mobile healthcare, which were indicated in data collection, data storage, data analytics and knowledge creation.

More information can be found in [1.6], [1.19], [1.33], [1.39] and in [1.55].

1.7. BIG DATA REAL-LIFE EXAMPLES

As we are heading towards the end of the introductory chapter of this book, it will be reasonable to emphasize some real-life examples of big data solutions, so that we may become familiar with even more interesting (than presented earlier) implementation of big data systems and become encouraged to further explore the topic.

Let us consider the following examples:

- improving time use measurement with big data [1.60];
- enabling big data at manufacturing fields of automotive company [1.1];
- application of big data for credit scoring [1.41];
- use of big data in libraries [1.40];
- analysis of campus life with big data approach [1.58];
- selecting Smart Village in India by use of big data analytics [1.45];
- use of big data in ocean's observation [1.32];
- use of big data in Health Care [1.18];
- use of big data in driver state monitoring [1.5].

Let us focus on the last real-life example. Driver's distraction, sleepiness or stress are identified as factors of car accidents. By using heterogeneous sensors, we obtain data from multiple sources and by use of big data solutions, they are processed and analyzed in order to inform vehicle drivers.

More interesting real-life examples can be found in [1.51] and [1.8].

1.8. SUMMARY

This first chapter was devoted to big data fundamentals. From both perspectives, theoretical and practical, we have presented a definition of big data, big data applications, big and big data security and privacy issues. Furthermore, we have discussed big data and ethics and big data challenges, as well as big data real-life examples.

Please go through the references itemized below and explore in more detail the chosen, interesting topic.

In the next chapter, we will prepare for work with big data.

1.9. EXERCISES

1. Have you come across a big data solution? If yes, please prepare its short description for further discussion.

2. Analyze definitions of big data provided in literature cited within the chapter. Which definition in your opinion is of the highest quality and why?
3. Identify characteristics of big data (5 V's) and illustrate them with examples based on the world of social media.
4. Prepare the mind map showing big data based on digital traces generated by user interactions using mobile phone applications.
5. Identify security aspects related with big data solutions. Define an asset (dataset), provide the context (e.g., e-bookstore) and then prepare the list of security risks and security countermeasures.
6. How does GDPR relate with big data? Explore the topic and prepare the short note about it.
7. Search the Internet and look for cases related with big data and ethics. Choose three of them and prepare their short description, for further academic discussion.
8. Go through a subsection devoted to big data challenges, pick one challenge and prepare the basis for assessment by compiling a list of strengths and weaknesses.
9. Conduct a thought experiment related to a big data solution prepared based on student interaction with the learning management system.
10. Prepare a presentation of a chosen example of real-life big data. Are there any other examples which are similar to the one that was chosen?

1.10. References

1.10.1. Bibliography

[1.1] Akin, Ozgun et al., *Enabling Big Data Analytics at Manufacturing Fields of Farplas Automotive*, arXiv:2004.11682 [cs], April 2020, arXiv.org, http://arxiv.org/abs/2004.11682.

[1.2] Almasaari, Shakir A., *Securing Big Data systems, A cybersecurity management discussion*, arXiv:1912.08191 [cs], December 2019, arXiv.org, http://arxiv.org/abs/1912.08191.

[1.3] Aziz, Fayeem et al., *Big Data in IoT Systems*, arXiv:1905.00490 [cs], April 2019, arXiv.org, http://arxiv.org/abs/1905.00490.

[1.4] Azmoodeh A., Dehghantanha A. (2020), *Big Data and Privacy: Challenges and Opportunities*, In: Choo K.K., Dehghantanha A. (eds.), *Handbook of Big Data Privacy*, Springer, Cha.

[1.5] Barua S., Begum S., Ahmed M.U. (2016), *Driver's State Monitoring: A Case Study on Big Data Analytics*, In: Ahmed M., Begum S., Raad W. (eds.), *Internet of Things Technologies for HealthCare*, HealthyIoT 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 187, Springer, Cham.

[1.6] Bohlouli, Mahdi et al., *Towards an Integrated Platform for Big Data Analysis*, arXiv:2004.13021 [cs], April 2020, arXiv.org, http://arxiv.org/abs/2004.13021.

[1.7] Che D., Safran M., Peng Z. (2013), *From Big Data to Big Data Mining: Challenges, Issues, and Opportunities*, In: Hong B., Meng X., Chen L., Winiwarter W., Song W. (eds.), *Database Systems for Advanced Applications*, DASFAA 2013. Lecture Notes in Computer Science, vol 7827, Springer, Berlin, Heidelberg.

[1.8] Chebbi I., Boulila W., Farah I.R. (2015), *Big Data: Concepts, Challenges and Applications*, In: Núñez M., Nguyen N., Camacho D., Trawiński B. (eds.), *Computational Collective Intelligence*, Lecture Notes in Computer Science, vol 9330, Springer, Cham.

[1.9] Cheng X., Fang L., Yang L., Cui S. (2018), *Mobile Big Data*, In: *Mobile Big Data*, Wireless Networks, Springer, Cham.

[1.10] Christen M., Blumer H., Hauser C., Huppenbauer M. (2019), *The Ethics of Big Data Applications in the Consumer Sector*, In: Braschler M., Stadelmann T., Stockinger K. (eds.), *Applied Data Science*, Springer, Cham.

[1.11] Cremonesi M., Bellini C., Bian B., Canali L., Dimakopoulos V., Elmer P., Fisk I., Girone M., Gutsche O., Hoh S.Y. et al. (2019), *Using Big Data Technologies for HEP Analysis*, EPJ Web of Conferences, 214, 06030.

[1.12] Curry E. (2016), *The Big Data Value Chain: Definitions, Concepts, and Theoretical Approaches*, In: Cavanillas J., Curry E., Wahlster W. (eds.), *New Horizons for a Data-Driven Economy*, Springer, Cham.

[1.13] Dai H.N., Wang H., Xu G., Wan J., Imran M. (2019), *Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies*, Enterprise Information Systems, 1-25.

[1.14] D'Alconzo A., Drago I., Morichetta A., Mellia M., Casas P. (2019), *A Survey on Big Data for Network Traffic Monitoring and Analysis*, IEEE Transactions on Network and Service Management, 16(3), 800-813.

[1.15] Demchenko Y., Ngo C., de Laat C., Membrey P., Gordijenko D. (2014), *Big Security for Big Data: Addressing Security Challenges for the Big Data Infrastructure*, In: Jonker W., Petković

M. (eds.), *Secure Data Management*, SDM 2013. Lecture Notes in Computer Science, vol 8425, Springer, Cham.

[1.16] Edward S.G., Sabharwal N. (2015), *Big Data*, In: *Practical MongoDB*, Apress, Berkeley, CA.

[1.17] Elgendy N., Elragal A. (2014), *Big Data Analytics: A Literature Review Paper*, In: Perner P. (eds.), *Advances in Data Mining. Applications and Theoretical Aspects*, ICDM 2014. Lecture Notes in Computer Science, vol 8557, Springer, Cham.

[1.18] Gaitanou P., Garoufallou E., Balatsoukas P. (2014), *The Effectiveness of Big Data in Health Care: A Systematic Review*, In: Closs S., Studer R., Garoufallou E., Sicilia M.A. (eds.), *Metadata and Semantics Research*, MTSR 2014. Communications in Computer and Information Science, vol 478, Springer, Cham.

[1.19] Gorodetsky V. (2014), *Big Data: Opportunities, Challenges and Solutions*, In: Ermolayev V., Mayr H., Nikitchenko M., Spivakovsky A., Zholtkevych G. (eds.), *Information and Communication Technologies in Education, Research, and Industrial Applications*, ICTERI 2014. Communications in Computer and Information Science, vol 469, Springer, Cham.

[1.20] Gronwald K.D. (2017), *Business Intelligence (BI) and Big Data Analytics (Big Data)*, In: *Integrated Business Information Systems*, Springer, Berlin, Heidelberg.

[1.21] Guo L., Xu W., Li H., Zhang S., Zhao D. (2016), *The Application of Big Data Technology in the Field of Combat Simulation Data Management*, In: Zhang L., Song X., Wu Y. (eds.), *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems*, AsiaSim 2016, SCS Autumn Sim 2016. Communications in Computer and Information Science, vol 645, Springer, Singapore.

[1.22] Helbing D. (2015), *Big Data – A Powerful New Resource for the Twenty-first Century*, In: *Thinking Ahead – Essays on Big Data, Digital Revolution, and Participatory Market Society*, Springer, Cham.

[1.23] Hou W., Guo P., Guo L. (2015), *Networking Big Data: Definition, Key Technologies and Challenging Issues of Transmission*, In: Wang Y., Xiong H., Argamon S., Li X., Li J. (eds.), *Big Data Computing and Communications*, BigCom 2015. Lecture Notes in Computer Science, vol 9196, Springer, Cham.

[1.24] Hussein E., Sadiki R., Jafta Y., Sungay M.M., Ajayi O., Bagula A. (2020), *Big Data Processing Using Hadoop and Spark: The Case of Meteorology Data*, In: Zitouni R., Agueh M., Houngue P., Soude H. (eds.), *e-Infrastructure and e-Services for Developing Countries*, AFRICOMM 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 311, Springer, Cham.

[1.25] Jaraba Navas P.C., Guacaneme Parra Y.C., Rodríguez Molano J.I. (2016), *Big Data Tools: Hadoop, MongoDB and Weka*, In: Tan Y., Shi Y. (eds.), *Data Mining and Big Data*, DMBD 2016. Lecture Notes in Computer Science, vol 9714, Springer, Cham.

[1.26] Jedlitschka A. (2017), *Analyzing the Potential of Big Data*, In: Felderer M., Méndez Fernández D., Turhan B., Kalinowski M., Sarro F., Winkler D. (eds.), *Product-Focused Software Process Improvement*, PROFES 2017. Lecture Notes in Computer Science, vol 10611, Springer, Cham.

[1.27] Khanan A., Abdullah S., Mohamed A.H.H.M., Mehmood A., Ariffin K.A.Z. (2019), *Big Data Security and Privacy Concerns: A Review*, In: Al-Masri A., Curran K. (eds.), *Smart Technologies and Innovation for a Sustainable Future*, Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development), Springer, Cham.

[1.28] Lake P., Crowther P. (2013), *Big Data*, In: *Concise Guide to Databases*, Undergraduate Topics in Computer Science, Springer, London.

[1.29] Lake P., Drake R. (2014), *Introducing Big Data*, In: *Information Systems Management in the Big Data Era*, Advanced Information and Knowledge Processing, Springer, Cham.

[1.30] Lee N. (2014), *Consumer Privacy in the Age of Big Data*, In: *Facebook Nation*, Springer, New York, NY.

[1.31] Li Q. et al. (2019), *Big Data Architecture and Reference Models*, In: Debruyne C., Panetto H., Guédria W., Bollen P., Ciuciu I., Meersman R. (eds.), *On the Move to Meaningful Internet Systems: OTM 2018 Workshops*, OTM 2018. Lecture Notes in Computer Science, vol 11231, Springer, Cham.

[1.32] Liu Y., Qiu M., Liu C., Guo Z. (2016), *Big Data in Ocean Observation: Opportunities and Challenges*, In: Wang Y., Yu G., Zhang Y., Han Z., Wang G. (eds.), *Big Data Computing and Communications*, BigCom 2016. Lecture Notes in Computer Science, vol 9784, Springer, Cham.

[1.33] Lyu F., Ren L., Du Y. (2017), *An Optimization Method for User Interface Components Based on Big Data*, In: Zhang L., Ren L., Kordon F. (eds.), *Challenges and Opportunity with Big Data*, Monterey Workshop 2016. Lecture Notes in Computer Science, vol 10228, Springer, Cham.

[1.34] Mazumder S. (2016), *Big Data Tools and Platforms*, In: Yu S., Guo S. (eds.), *Big Data Concepts, Theories, and Applications*, Springer, Cham.

[1.35] Mikalef P., Pappas I.O., Krogstie J. et al. (2018), *Big data analytics capabilities: a systematic literature review and research agenda*, Inf Syst E-Bus Manage 16, 547-578.

[1.36] Mohanty S., Jagadeesh M., Srivatsa H. (2013), *Application Architectures for Big Data and Analytics*, In: *Big Data Imperatives*, Apress, Berkeley, CA.

[1.37] Nafchi, Mohsen Aghabozorgi, Maryam Aghabozorgi Nafchi, *Challenges and Opportunities of Big Data in Healthcare Mobile Applications*, arXiv:1906.10166 [cs], June 2019, arXiv.org, http://arxiv.org/abs/1906.10166.

[1.38] Nataliya, Shakhovska et al., *Generalized formal model of big data*, arXiv:1905.03061 [cs], May 2019, arXiv.org, http://arxiv.org/abs/1905.03061.

[1.39] Obitko M., Jirkovský V., Bezdíček J. (2013), *Big Data Challenges in Industrial Automation*, In: Mařík V., Lastra J.L.M., Skobelev P. (eds.), *Industrial Applications of Holonic and Multi-Agent Systems*, Lecture Notes in Computer Science, vol 8062, Springer, Berlin, Heidelberg.

[1.40] Olendorf R., Wang Y. (2017), *Big Data in Libraries*, In: Suh S., Anthony T. (eds.), *Big Data and Visual Analytics*, Springer, Cham.

[1.41] Óskarsdóttir M., Bravo C., Sarraute C., Vanthienen J., Baesens B. (2019), *The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics*, Applied Soft Computing, 74, 26-39.

[1.42] Pastor-Escuredo, David, *Ethics in the digital era*, arXiv:2003.06530 [cs], March 2020, arXiv.org, http://arxiv.org/abs/2003.06530.

[1.43] Pormeister K. (2017), *The GDPR and Big Data: Leading the Way for Big Genetic Data?*, In: Schweighofer E., Leitold H., Mitrakas A., Rannenberg K. (eds.), *Privacy Technologies and Policy*, APF 2017. Lecture Notes in Computer Science, vol 10518, Springer, Cham.

[1.44] Prabhu C., Chivukula A., Mogadala A., Ghosh R., Livingston L. (2019), *Big Data Analytics in Bio-informatics*, In: *Big Data Analytics: Systems, Algorithms, Applications*, Springer, Singapore.

[1.45] Radhika D., Aruna Kumari D. (2018), *Adding Big Value to Big Businesses: A Present State of the Art of Big Data, Frameworks and Algorithms*, In: Saini A., Nayak A., Vyas R. (eds.), *ICT Based Innovations*, Advances in Intelligent Systems and Computing, vol 653, Springer, Singapore.

[1.46] Rafferty W., Rafferty L., Hung P.C.K. (2016), *Introduction to Big Data*, In: Hung P. (eds.), *Big Data Applications and Use Cases*, International Series on Computer Entertainment and Media Technology, Springer, Cham.

[1.47] Rambousek A., Parkin H., Horak A. (2018), *Software Tools for Big Data Resources in Family Names Dictionaries Names*, 66(4), 246-255.

[1.48] Ramya Devi R., Vijaya Chamundeeswari V., *Triple DES: Privacy Preserving in Big Data Healthcare*, Int J Parallel Prog 48, 515-533 (2020), https://doi.org/10.1007/s10766-018-0592-8.

[1.49] Rosenfeld, Ariel et al., *Big Data Analytics and AI in Mental Healthcare*, arXiv:1903.12071 [cs], March 2019, arXiv.org, http://arxiv.org/abs/1903.12071.

[1.50] Sangeetha S., Sudha Sadasivam G. (2019), *Privacy of Big Data: A Review*, In: Dehghantanha A., Choo K.K. (eds.), *Handbook of Big Data and IoT Security*, Springer, Cham.

[1.51] Santos A.F.C., Teles Í.P., Siqueira O.M.P., de Oliveira A.A. (2018), *Big Data: A Systematic Review*, In: Latifi S. (eds.), *Information Technology – New Generations*, Advances in Intelligent Systems and Computing, vol 558, Springer, Cham.

[1.52] Shi Y., Quan P. (2020), *Big Data Analysis: Theory and Applications*, In: Lirkov I., Margenov S. (eds.), *Large-Scale Scientific Computing*, LSSC 2019. Lecture Notes in Computer Science, vol 11958, Springer, Cham.

[1.53] Spraker K. (2018), *Difficulties Implementing Big Data: A Big Data Implementation Study*, In: Kurosu M. (eds.), *Human-Computer Interaction*, Interaction in Context. HCI 2018. Lecture Notes in Computer Science, vol 10902, Springer, Cham.

[1.54] Steinmann M. et al. (2015), *Embedding Privacy and Ethical Values in Big Data Technology*, In: Matei S., Russell M., Bertino E. (eds.), *Transparency in Social Media*, Computational Social Sciences, Springer, Cham.

[1.55] Trevisan, Martino, *Big Data for Traffic Monitoring and Management*, arXiv:1902.11095 [cs], February 2019, arXiv.org, http://arxiv.org/abs/1902.11095.

[1.56] Wani M.A., Jabin S. (2018), *Big Data: Issues, Challenges, and Techniques in Business Intelligence*, In: Aggarwal V., Bhatnagar V., Mishra D. (eds.), *Big Data Analytics*, Advances in Intelligent Systems and Computing, vol 654, Springer, Singapore.

[1.57] Whang K.Y. (2018), *Recent Trends of Big Data Platforms and Applications*, In: Trujillo J. et al. (eds.), *Conceptual Modeling*, ER 2018. Lecture Notes in Computer Science, vol 11157, Springer, Cham.

[1.58] Yang, Zongkai et al., *Evolution Features and Behavior Characters of Friendship Networks on Campus Life*, arXiv:2004.06266 [physics, stat], April 2020, arXiv.org, http://arxiv.org/abs/2004.06266.

[1.59] Yue Y., Li D. (2019), *Big Data Challenges of FAST*, Lecture Notes in Computer Science, 6-9.

[1.60] Zeni, Mattia et al., *Improving time use measurement with personal big data collection – the experience of the European Big Data Hackathon 2019*, arXiv:2004.11940 [cs], April 2020, arXiv.org, http://arxiv.org/abs/2004.11940.

[1.61] Zhu, Dingju, *Big Data based Research on Mechanisms of Sharing Economy Restructuring the World*, arXiv:2001.08926 [econ, q-fin], January 2020, arXiv.org, http://arxiv.org/abs/2001.08926.

1.10.2. INTERNET REFERENCES

[1.62] https://en.wikipedia.org/wiki/Big_data

[1.63] https://www.allaccess.com/merge/archive/31294/infographic-what-happens-in-an-internet-minute

Chapter 2. Preparation to work with big data

Thanks to big data, machines can now be programmed to the next thing right.
But only humans can do the next right thing.

D. Seidman

2.1. INTRODUCTION

In this chapter, we will provide a primer aimed at preparation to work with large data sets. According to [2.18], [2.4] big data ecosystem is extremely complex, so we will discuss it in general without going into details. Nevertheless, it is recommended to pick the interesting element of the big data ecosystem and acquire more information about it.

Before we move to the practical aspects of big data, we will introduce the reader with opening examples as well as we will focus on dataset sources (subchapter 2.2), dataset processing and visualization (subchapter 2.3). We assume that starting from examples which are based on *small data* will prepare a good starting point for further presentation of issues related to *big data*.

We strongly encourage you to explore the references section. It is worth to be noticed, that included references individually which will be deepened by the reader at his own pace, could be very valuable.

In the final section of this chapter, a list of exercises is provided.

That list can be useful for both purposes: to indicate the most important topics covered in this chapter and to serve as a teaching aid for classes in the subject related to big data.

2.2. DATA VISUALIZATION

Let us begin with something that is very close to us, as users of any information system. We start with the output, which can be for example: a number, a series of numbers, a character or a series of characters or it can contain image, images or maybe multimedia. It can be non-interactive or it can offer controls which will allow it to shape the output.

In the further part of this subsection, let us assume we will focus on three chosen examples of data visualization.

Example number one is based on a famous source entitled "*Information is beautiful*" [2.24]. In Figure 2.1 we can see the screenshot documenting World's biggest data breaches.

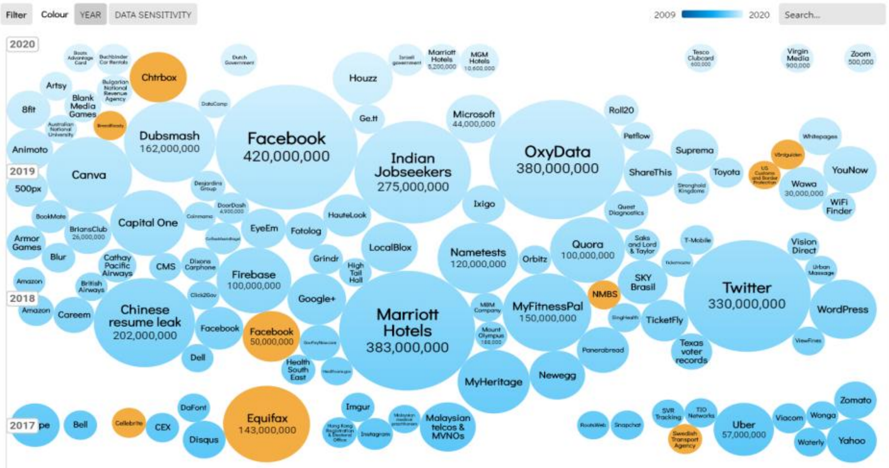


Figure 2.1. World's biggest data breaches. *Source: https://bit.ly/bigdata2020-isb*

We can apply the filter to choose the sector (e.g., academic, government, healthcare, media, military, retail, telecoms, transport, web) or the method (hacked, inside job, lost device, etc.) or define what does the color represent (by default it represents year, but it can be changed to data sensitivity). There is a search field as well as the possibility to download the source data.

The first example inspires the questions about the data source and about the tools used to prepare such visualization. In this case, the data source was prepared by project team members and the tool is an internal tool called VIZSweet. There are other visualizations, like: "Snake Oil Supplements" or "Common MythConceptions", we encourage you to explore the examples by vitising: https://informationisbeautiful.net/visualizations.

Example number 2 refers to data visualization created as part of the project called "Natabilia" (http://notabilia.net). This project is aimed at visualizing deletion discussions on Wikipedia. The story of an article begins with submission of the first version of the article, which is reviewed by the community, to make sure that the article is compliant with Wikipedia's guidelines. An article can be nominated for deletion, after which the discussion takes place and community members are discussing in favor or against keeping the given article on Wikipedia. Thanks to this project it is possible to look into the collective decisions process undergoing in creating Wikipedia's content.

As we can see in Figure 2.2, there is a tree-like visualization representing the top 100 longest discussions, which ended in the deletion of a given article (the random discussion concerning "Children of Michael Jackson" was selected). Please notice that the green color represents a positive decision ("keep") and the red color represents a negative decision ("delete").

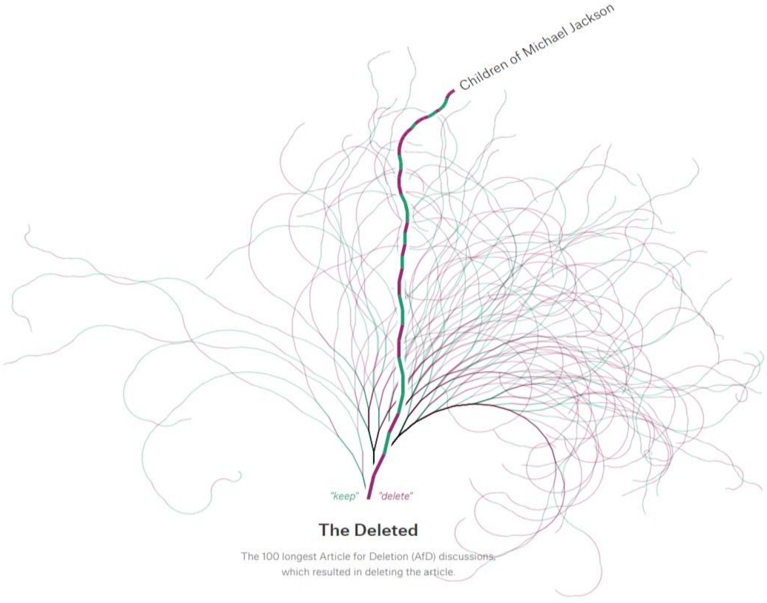


Figure 2.2. Notabilia. *Source: https://bit.ly/bigdata2020-notabilia*

Exploration of this project leads to the identification of *Article for Deletion* patterns. Among them, there are: "controversial" (opposing opinions balance each other), "swinging" (series of "keeps" is followed by series of "deletes") or "unanimous" (participants represent total agreement). As we can read on the website of the project, the analysis of a large dataset of *Article for Deletion* discussions (more than 200 000 discussions) suggests that the largest part of discussions ends after the expression of just a few recommendations.

More detailed analysis (e.g., how hard to reach the consensus) and more interesting facts (e.g., how quickly new participants are joining the discussion) are presented on the project's website and exploring them gives a chance to have a complete view of obtained results.

Another opening example (the third one) is related to visual literacy. On a website located at: https://www.visual-literacy.org, we can find interactive knowledge maps and among them, there is a periodic table of visualization methods (see Fig. 2.3).

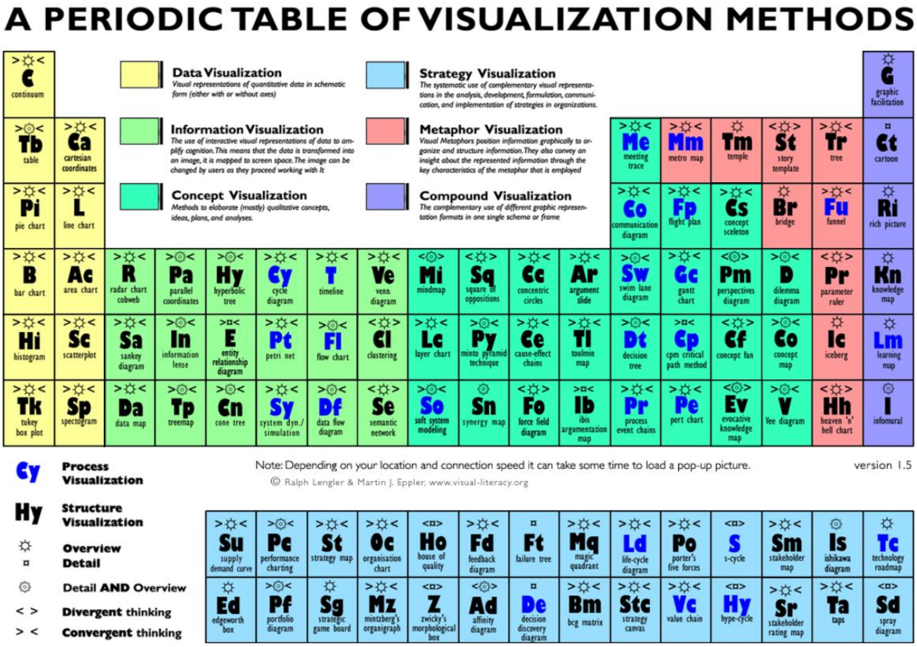


Figure 2.3. A periodic table of visualization methods.
Source: https://bit.ly/bigdata2020-periodic-interactive

Different colors represent different categories of visualization methods:

a) **data visualization**, e.g.: table, pie chart, line chart, area chart, bar chart, histogram, scatterplot, spectrogram),
b) **information visualization**, e.g.: treemap, timeline, data flow diagram,
c) **concept visualization**, e.g.: mindmap, layer chart, decision tree, dilemma diagram,

d) **strategy visualization**, e.g.: failure tree, technology roadmap; BCG matrix, value chain, strategy map,
e) **metaphor visualization**, e.g.: tree, funnel, iceberg, temple, bridge,
f) **compound visualization**, e.g.: cartoon, rich picture, knowledge map.

It is worth mentioning that we can get detailed information about each element, by hovering the mouse cursor over it. For further information about that project, please read the paper entitled "*Towards A Periodic Table of Visualization Methods for Management*" which can be downloaded from https://bit.ly/bigdata2020-periodic.

Another fascinating project called "*DataViz Project*" shows the result of bringing "*all possible*" data visualization on its portal, which is located at: https://datavizproject.com/.

We can narrow the list of data visualizations by selecting the family, input, function or shape.

First, we will enumerate the data visualizations, by categorizing them into:

a) Charts (C.x),
b) Diagrams (D.x),
c) Other (O.x).

The list of charts consists of (presented in alphabetical order):

- C01. **Area chart**. It is a line graph with the area below filled with colors;
- C02. **Bar chart (horizontal)**. It is a visual presentation of categorical data by the use of rectangular bars;
- C03. **Bar chart (vertical)**. It is a chart with rectangular bars;
- C04. **Bar chart on a map**. It is a combination of a map and a bar chart;
- C05. **Bubble chart**. It is a scatter plot using bubbles instead of data points;
- C06. **Bubble map chart**. It is a combination of a bubble chart and a map;
- C07. **Bump chart**. It illustrates the changes in rank over time;
- C08. **Butterfly chart**. It shows two sets of data series side by side;
- C09. **Candlestick chart**. It is used to show price movements;
- C10. **Comparison chart**. It consists of rows and columns prepared for comparison purposes (it has a spreadsheet structure);
- C11. **Compound bubble and pie chart**. It shows the performance across four parameter sets;
- C12. **Curved bar chart**. It is a variation of a bar chart using curved areas instead of bars;
- C13. **Donut chart**. It is a pie chart with a blank center;
- C14. **Fan chart (genealogy)**. It is a representation of family relationships in a tree structure;
- C15. **Fan chart (time series)**. It is a chart that combines line graphs (for representation of the past) and range area charts (for representation of future predictions);
- C16. **Flow chart**. It is a visual representation of processes or workflow by boxes and arrows;

- C17. **Funnel chart**. It shows streamlined data by displaying values as progressively decreasing proportions;
- C18. **Gantt chart**. It shows tasks or events displayed against time (used in project management);
- C19. **Group bar chart**. It is a simple bar chart with at least two graphs grouped under specified categories;
- C20. **Kagi chart**. It is a chart without a time axis built from a series of vertical and horizontal lines;
- C21. **Layered area chart**. It is a multiple area chart with layers represented by the use of perspective (or transparency);
- C22. **Lollipop chart**. This chart is similar to a bar chart but instead of using bars it uses vertical lines with dots at the end;
- C23. **Marimekko chart**. It is a 2D stacked chart;
- C24. **Multi-level donut chart**. It is a set of concentric circles used to visualize hierarchical relationships;
- C25. **Multi-level pie chart**. This chart includes tree structures in a pie chart);
- C26. **Multiple series 3D bar charts**. It is a visualization of multiple datasets by the use of 3D bars;
- C27. **Nested proportional area chart**. This chart is built using bubbles allowing comparison of proportions;
- C28. **Organizational chart**. It shows the structure of a given organization;
- C29. **Packed circle chart**. This chart is composed of circles used to visualize hierarchically structured data;
- C30. **Pareto chart**. This chart consists of bars (presented in descending order) and a line graph that shows the cumulative total;
- C31. **Partition layer chart**. It is a visual representation of clustering results;
- C32. **Pictorial bar chart**. It is a variation of a bar chart using icons instead of bars;
- C33. **Pictorial fraction chart**. It is a chart showing fractions by using pictograms (icons, pictures, etc.);
- C34. **Pictorial stacked chart**. It is a chart with a pictogram;
- C35. **Pictorial unit chart**. It is a visualization of data by use of icons, symbols or pictures;
- C36. **Pie chart**. It is a chart in the form of a circle divided into sectors according to numerical proportion;
- C37. **Pie chart map**. It is a combination of pie chart and a map;
- C38. **Polar area chart**. It is a pie chart with sectors of equal angles and different lengths from the center;
- C39. **Polar chart**. It shows the multivariate data in the form of a 2D chart;
- C40. **Proportional area chart (circle)**. It is a chart used for proportions comparison (size of data is expressed by circles);
- C41. **Proportional area chart (halfcircle)**. It is a variation of a proportional area chart with one measure depicted as a circle;
- C42. **Proportional area chart (icon)**. It is used for comparing proportions and using icons of different sizes;

- C43. **Proportional area chart (square)**. It is a chart with squares of different sizes allowing comparison of proportions;
- C44. **Pyramid chart**. It is an inverted funnel chart;
- C45. **Radial area chart**. It is a modified area chart with a radial basis;
- C46. **Radial bar chart**. It is a bar chart which is displayed on a polar coordinate system;
- C47. **Range area chart**. It is a variation of area charts allowing plot bands of data (e.g., weather patterns);
- C48. **Renko chart**. It shows trends over time (for example price changes);
- C49. **Semi circle donut chart**. It is a half donut chart;
- C50. **Slope chart**. It is a line chart with specified exactly two points in time;
- C51. **Solid gauge chart**. It is a simplified angular gauge chart showing whether something is good or bad;
- C52. **Span chart**. It shows a range of data by plotting two Y values per given data point;
- C53. **Stacked area chart**. It is a simple area chart, but using multiple data series;
- C54. **Stacked bar chart**. This chart shows how the larger category is divided into smaller ones;
- C55. **Stacked ordered area chart**. It shows the change of order over time;
- C56. **Swimlane flowchart**. It shows steps of given processes with showing to whom a particular step belongs;
- C57. **Table chart**. It is a representation of data in rows and columns;
- C58. **Tally chart**. It is a graphical method using a tally mark numeric system;
- C59. **Triangle bar chart**. It is a variation of a bar chart using triangles instead of rectangles;
- C60. **Waffle chart**. It is a grid of small cells which are colored proportionally to progress;
- C61. **Waterfall chart**. It shows cumulative results of positive and negative values.

Let us take a look at some visualizations, for example: bump chart, C07 (Fig. 2.4) and funnel chart, C17 (Fig. 2.5).

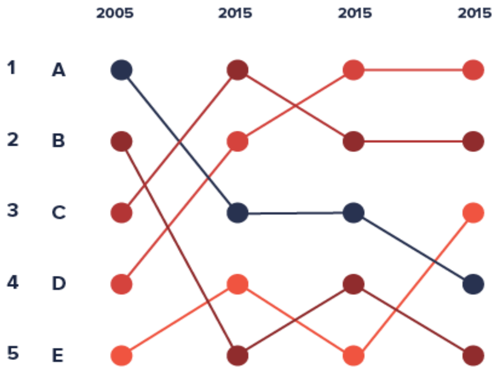


Figure 2.4. An example of bump chart, *Source: https://bit.ly/bigdata2020-bump*

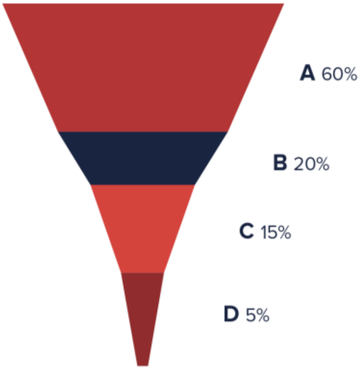


Figure 2.5. An example of funnel chart. *Source: https://bit.ly/bigata2020-funnel*

The list of diagrams consists of (presented in alphabetical order):

- D01. **Alluvial flow diagram**. It represents changes in network structure over time;
- D02. **Arc diagram**. It is a diagram with nodes and arcs which are representing connections;
- D03. **Chord diagram**. This diagram shows arranged radial relationships between data;
- D04. **Cycle diagram**. This diagram shows repetitive events;
- D05. **Euler diagram**. It is a visual representation of sets and relationships between them;
- D06. **Fishbone diagram**. It is a causal diagram showing causes of a given outcome;
- D07. **Illustration diagram**. It is an image with labels, notes and a legend, prepared for explanation purposes;
- D08. **Linear process diagram**. It is a visualization of a process by use of a set of connected points representing process steps;
- D09. **Matrix diagram (Y-shaped)**. It is a variation of matrix diagram;

- D10. **Matrix diagram roof shaped**. It is a matrix diagram showing relationships between items;
- D11. **Matrix diagram**. It shows the relationship which can be present or absent at a given intersection between items;
- D12. **Molecule diagram**. It is a visual representation of the molecule;
- D13. **Non-ribbon chord diagram**. It is a modified version of a chord diagram showing interrelationships between data;
- D14. **Opposite diagram**. In this diagram, data points are presented using Cartesian coordinates with two set off opposites (e.g., good, bad; simple, complex);
- D15. **Phase diagram**. It shows the conditions under which distinct phases can occur at equilibrium (used for example in physical chemistry);
- D16. **Process diagram**. It shows the process steps;
- D17. **Pyramid diagram**. It is a diagram used to show hierarchical structure;
- D18. **Radar diagram**. It is a 2D chart of three (or more) variables represented on axes starting from the central point;
- D19. **Sankey diagram**. It is a flow diagram with arrows proportional to flow quantity;
- D20. **Sunburst diagram**. It is a diagram with concentric circles representing hierarchical data;
- D21. **Target diagram**. It is a layered diagram with a goal at the center that displays progress towards that goal;
- D21. **Taylor diagram**. It is a graphical way to show how a given pattern is close to observations;
- D22. **Venn diagram**. It shows logical relations between sets.

Let us provide some examples of visualizations of two chosen diagrams, namely: D21 (Taylor diagram) presented in Figure 2.6 and D14 (Opposite diagram), presented in Figure 2.7.

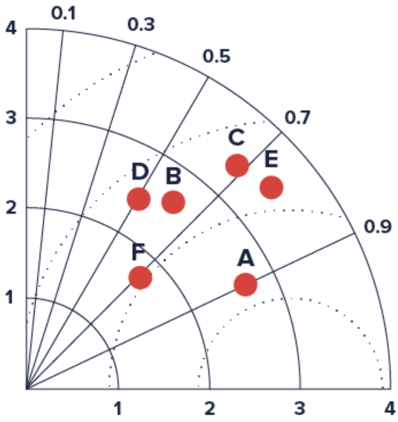


Figure 2.6. An example of Taylor diagram. *Source: https://bit.ly/bigdata2020-taylor*

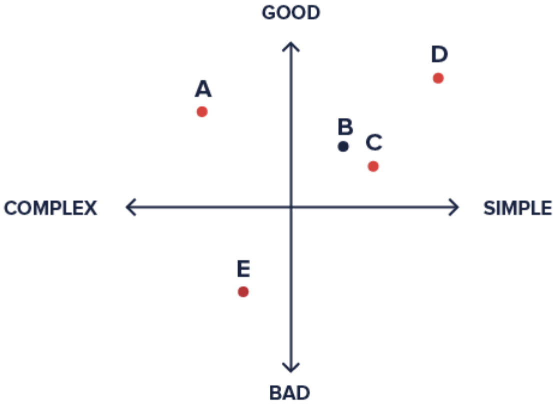


Figure 2.7. An example of Opposite diagram. *Source: https://bit.ly/bigdata2020-opposite*

The list of other (not diagrams or charts) data visualizations consists of (presented in alphabetical order):

- O01. **3D scatter plot**. It is a scatter plot with three variables;
- O02. **Angular gauge**. It is a chart which uses a radial scale (looking like a speedometer);
- O03. **Bagplot**. It is a variation of box plots useful for visualizing two-dimensional data;
- O04. **Boxplot**. It shows the groups of data through their quartiles;
- O05. **Bubble timeline**. It is a timeline combined with variables displayed by the use of bubbles of different sizes;
- O06. **Bullet graph**. It is a bar graph inspired by a classic thermometer or progress bar;
- O07. **Cartogram**. It uses distorted shapes of geographic regions;
- O08. **Choropleth map**. It is a map with areas shaded in proportion to the variable displayed on the map (e.g., population density);
- O09. **Circular heat map**. It is a heatmap which is aligned radially;
- O10. **Cluster analysis**. It is based on an assumption of grouping similar objects in order to create clusters;
- O11. **Clustered force layout**. It is a grouped bubble chart allowing to represent hierarchies;
- O12. **Column range**. It displays a range of data (expressed by upper and lower bounds of a given column;
- O13. **Column sparkline**. It is a sparkline using bars instead of lines;
- O14. **Connection map**. It displays networks combined with geographical data (e.g., for showing flight connections);
- O15. **Contour plot**. It shows in 2D the relationships among three numeric variables;
- O16. **Convex treemap**. It is a treemap using convex polygons (instead of rectangles);

- O17. **Dendrogram**. It is a tree diagram often used to illustrate the arrangement of the clusters;
- O18. **Development & causes**. It is a combination of a scaled timeline, area chart and line graph;
- O19. **Dot density map**. It is a map with dots representing a given phenomenon;
- O20. **Dot plot**. A chart consisting of data points represented as dots;
- O21. **Dumbbell plot**. It is a plot with dot plots built from two or more series of data;
- O22. **Exploded view drawing**. It is a diagram showing the order of assembly of various components of a given object;
- O23. **Flow map**. It is a mix of a map (or maps) and Sankey diagram;
- O24. **Hanging rootogram**. It combines theoretical curves with observed results;
- O25. **Heat map**. It is a way of visualization of values by use of variations in coloring;
- O26. **Hexagonal binning**. It is used to plot the density. Given data points are binned into gridded hexagons;
- O27. **Histogram**. It groups numeric data into bins in order to show the distribution of a given dataset;
- O28. **Hive plot**. It is a plot with radially distributed axes which can be used to draw large networks;
- O29. **Hyperbolic tree**. It is hierarchical data displayed as a tree composed of nodes and connections;
- O30. **Icon + number**. It is single data with an icon as a description of what the data is about;
- O31. **Icon count**. It is a combination of an icon and a number;
- O32. **Isoline map**. It shows data as a third dimension on the map;
- O33. **Line graph**. It is a series of data points connected by a straight line;
- O34. **Mind map**. It is a brainstorming map with a center, nodes and connections;
- O35. **Network visualization**. It visualizes relationships between a large number of entities (nodes);
- O36. **Parallel coordinates**. It is a plot using parallel axes allowing visualization of multivariate data;
- O37. **Parallel sets**. It is using boxes which are representing categories and parallelograms which are showing relationships between categories;
- O38. **Pin map**. It is a map with pins (representing geospatial data);
- O39. **Population pyramid**. It is a visualization showing the distribution of various age groups in a given population;
- O40. **Progress bar**. It is a graphical representation of the progression (e.g., file transfer);
- O41. **Radial convergences**. It is a visualization of relationships between entities;
- O42. **Radial histogram**. It is a histogram wrapped around a circle;
- O43. **Radial line graph**. It is a radial graph with data points wrapped around a circle;
- O44. **Route map**. It is a visualization of traces;
- O45. **Scaled timeline**. It is used to communicate time-related information;

- O46. **Scaled-up number**. It is used to emphasize a single value;
- O47. **Sociogram**. It is a visual representation of social relations;
- O48. **Sorted stream graph**. It is an area graph that is displaced around a central axis;
- O49. **Sparkline**. It is a small, simple graphic with typographic resolution;
- O50. **Spiral heat map**. It is a variation of a heat map prepared for comparable cycles;
- O51. **Spiral histogram**. It is a histogram with a timeline represented in a spiral shape;
- O52. **Spline graph**. It is a line graph that plots a curve through data points;
- O53. **Step by step illustration**. It is a series of pictures prepared to explain something (e.g., a process);
- O54. **Stepped line graph**. It is a chart with lines forming a series of steps between data points;
- O55. **Stream graph**. It is an area graph which is displaced around a central axis;
- O56. **SWOT analysis**. It is a method of visualization of: Strengths, Weaknesses, Opportunities and Threats of organization or a project;
- O57. **Ternary contour plot**. It is a variation of a ternary plot;
- O58. **Ternary plot**. It shows the ratios of the three variables as positions in a triangle;
- O59. **Three-dimensional stream graph**. It is a graph of a function (two variables) or a graph of the relationship (three variables);
- O60. **Timeline**. It is a visual representation of events represented in chronological order;
- O61. **Topographic map**. It is a visual representation of natural or cultural features on the ground;
- O62. **Transit map**. It is a topological map illustrating routes and stations;
- O63. **Treemap**. It displays data as a set of rectangles;
- O64. **Trendline**. It is a line showing the general course;
- O65. **Violin plot**. It is a box plot with a rotated kernel density plot;
- O66. **Waterfall plot**. It is a 3D plot in which curves of data are simultaneously displayed;
- O67. **Win-loss sparkline**. It is a sparkline showing only whether the value is positive or negative;
- O68. **Word cloud**. It is a visual representation of text with importance shown by the use of font size and color.

Let us bring some visual examples of the third (and last category) of data visualizations gathered within the DataViz *Project*: the population pyramid (Fig. 2.8) and an example of an opposite diagram (Fig. 2.9).

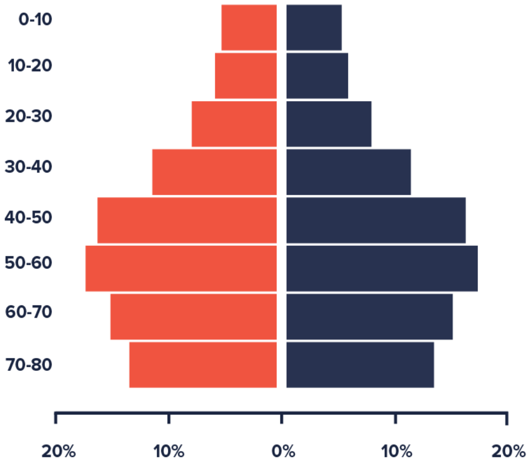


Figure 2.8. An example of population pyramid. *Source: https://bit.ly/bigdata2020-pyramid*

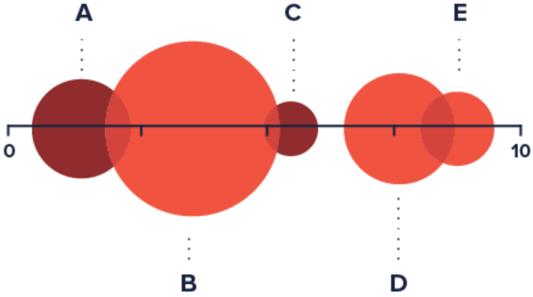


Figure 2.9. An example of opposite diagram. *Source: https://bit.ly/bigdata2020-bubble*

For more examples check the Internet references section (p. 2.11.2).

We have seen the potential related with the graphical representation of the data, obtained by use of data visualization tool (or tools) and of course, the data. Now, we will go further and bring practical aspects related with data and data processing.

2.3. Data sources

In Table 2.1 were gathered ten chosen sources of data which can be interesting from the point of view of our data science projects. Of course, that list is not a closed catalog, it is just a starting point. We will not describe each data source, instead, we will pick one data source from the list and perform a quick walkthrough.

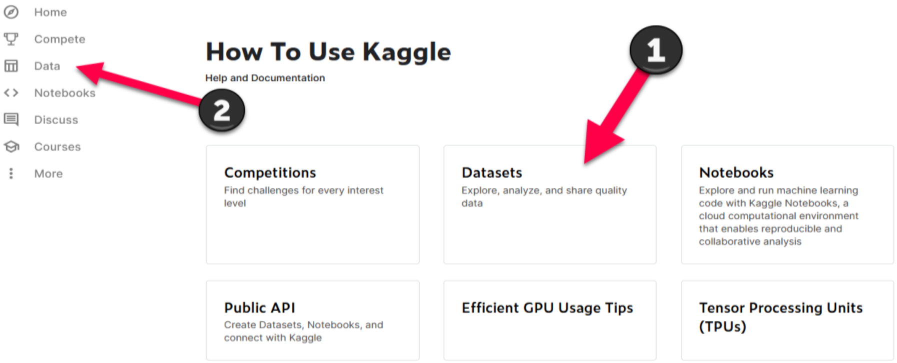
Tabable 2.1. Data sources

ID	URL
DS1	http://cocodataset.org
DS2	http://yann.lecun.com/exdb/mnist

DS3	https://catalog.data.gov/dataset
DS4	https://dane.gov.pl
DS5	https://data.mendeley.com
DS6	https://datacatalog.worldbank.org
DS7	https://datasetsearch.research.google.com
DS8	https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research
DS9	https://snap.stanford.edu/data
DS10	https://www.kaggle.com/datasets

Source: Author's own

We will elaborate on the data source with identifier DS10. Kaggle.com is an important Internet source of datasets which can be utilized in a data science project.



How To Use Kaggle

Help and Documentation

Competitions
Find challenges for every interest level

Datasets
Explore, analyze, and share quality data

Notebooks
Explore and run machine learning code with Kaggle Notebooks, a cloud computational environment that enables reproducible and collaborative analysis

Public API
Create Datasets, Notebooks, and connect with Kaggle

Efficient GPU Usage Tips

Tensor Processing Units (TPUs)

Figure 2.10. Document section of Kaggle portal (Kaggle.com).
Source: https://bit.ly/bigdata2020-kaggle

Fig. 2.10 shows the view of documentation provided by authors of the Kaggle portal, which has been divided into six parts: a) competitions, b) datasets, c) notebooks, d) public API, e) efficient GPU usage tips, and f) tensor processing units.

Since this subsection is about data sources, we will focus on two elements indicated in the above figure: (1) *Datasets documentation* and (2) *Data*.

Datasets documentation section (1) covers topics which are related to:

- types of datasets (with kind request to share datasets that are in non-proprietary format);
- supported file types (including: CSV, JSON, SQLite, archives and BigQuery);
- searching for dataset;
- creating a dataset;
- collaborating on datasets;
- resources for starting a data project;
- technical specifications (including dataset size limit, which is 20 GB).

If we explore the data section (2), we will have the possibility of *searching for a dataset* or *adding a new dataset*.

We will not present the process of adding a new dataset, but we will take a look at the search for the dataset.

We can use a search box or we can filter the list of datasets (see Fig. 2.11), by the specification of size, file type (1), license (2) or tags (3).

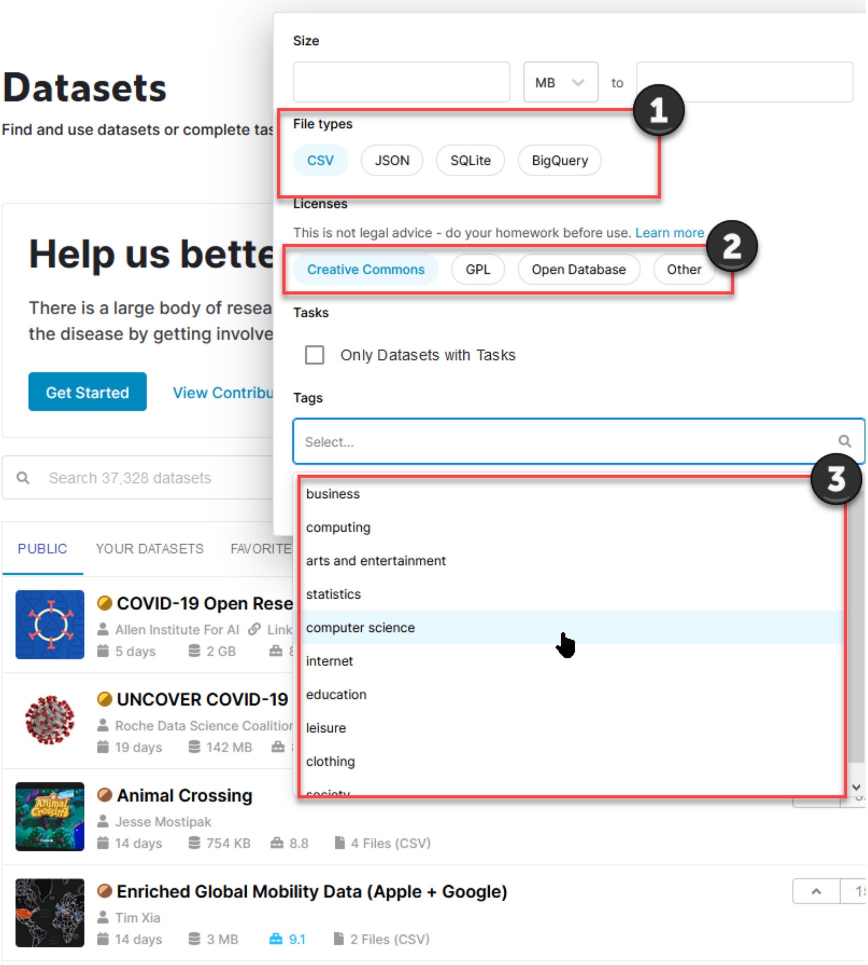


Figure 2.11. Searching for a dataset (Kaggle.com). *Source: https://bit.ly/bigdata2020-kaggle*

At the top of the list, there is a dataset entitled "*COVID-19 Open Research Dataset Challenge (CORD-19)*" with the following description (https://bit.ly/bigdata2020-covid): "*CORD-19 is a resource of over **63,000** scholarly articles, including over **51,000** with full text, about COVID-19, SARS-CoV-2, and related coronaviruses*" and the following call-to-action: "... *to develop text and data mining tools that can help the medical community develop answers to high priority scientific questions*".

Kaggle.com is just one of the exemplary data sources that can be used in data science projects, including those related with big data.

It is worth performing further steps, for example, make use of Kaggle Notebooks or participate in Kaggle Competitions. We assume that we are familiar with at least one source of data, which can be used in practical data science projects, which will utilize data tools.

2.4. DATA TOOLS

No doubt, there are many computer tools created to work with data, ranging from single-functionality applications to multi-functionality platforms. Instead of discussing the wide range of applications and platforms. Let us take a look at the chosen ten (see Tab. 2.2).

Table 2.2. Data science tools

ID	Name	URL
DT1	Google data studio	https://datastudio.google.com/
DT2	Jupyter	https://jupyter.org/
DT3	Knime	https://www.knime.com/
DT4	Microsoft PowerBI	https://powerbi.microsoft.com/
DT5	Octave	https://www.gnu.org/software/octave/
DT6	Orange	https://orange.biolab.si/
DT7	Qlik	https://www.qlik.com/
DT8	RapidMiner	https://rapidminer.com/
DT9	Tableau	https://www.tableau.com/products/cloud-bi
DT10	Weka3	https://www.cs.waikato.ac.nz/ml/weka/

Source: Author's own

Let us briefly describe just one example of a chosen data science tool, precisely DT6: *Orange Data Mining* (https://orange.biolab.si).

That computer tool is an open-source machine learning and data visualization platform, allowing to build data analysis workflows visually, by use of a predefined toolbox of widgets.

We start with a blank document and add a *file* element, as the data source. Next, we search or pick the appropriate widget (see Fig. 2.12). We navigate through the toolbox and in a visual way, build our data science solution.

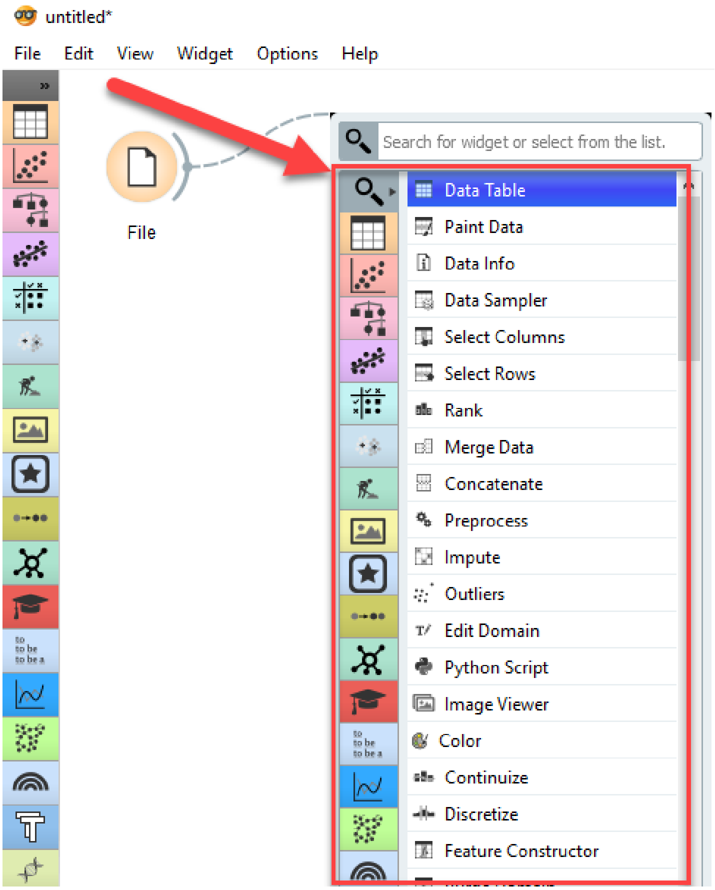
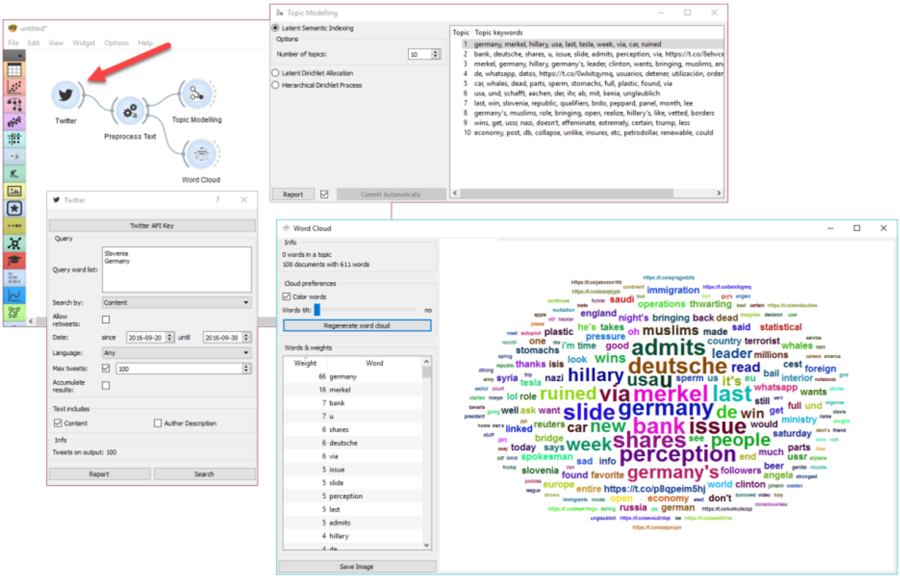


Figure 2.12. Orange Data Mining Tool (adding widget to the workflow).
Source: https://bit.ly/bigdata2020-orange

The solution enables exploratory data analysis of big data, thanks to SQL widget, which samples data. For example, Orange data mining tool makes it possible to create data science projects analyzing the social media data (see Fig. 2.13).



Twitter

Twitter API Key

Query

Query word list: Slovenia
Germany

Search by: Content

Allow retweets: □

Date: since 2016-09-20 until 2016-09-30

Language: Any

Max tweets: ☑ 100

Accumulate results: □

Text includes

☑ Content □ Author Description

Info

Tweets on output: 100

Report | Search

Topic Modelling

● Latent Semantic Indexing

Options

Number of topics: 10

○ Latent Dirichlet Allocation

○ Hierarchical Dirichlet Process

Report ☑ Commit Automatically

Topic	Topic keywords
1	germany, merkel, hillary, usa, last, tesla, week, via, car, ruined
2	bank, deutsche, shares, u, issue, slide, admits, perception, via, https://t.co/8ehvce
3	merkel, germany, hillary, germany's, leader, clinton, wants, bringing, muslims, an
4	de, whatsapp, datos, https://t.co/0wlxitqymq, usuarios, detener, utilización, order
5	car, whales, dead, parts, sperm, stomachs, full, plastic, found, via
6	usa, und, schafft, aachen, der, ihr, ab, mit, kenia, unglaublich
7	last, win, slovenia, republic, qualifiers, brdo, peppard, panel, month, lee
8	germany's, muslims, role, bringing, open, realize, hillary's, like, vetted, borders
9	wins, get, ussr, nazi, doesn't, effeminate, extremely, certain, trump, less
10	economy, post, db, collapse, unlike, insures, etc, petrodollar, renewable, could

Word Cloud

Info

0 words in a topic
100 documents with 611 words

Cloud preferences

☑ Color words

Words tilt: no

Regenerate word cloud

Words & weights

Weight	Word
66	germany
16	merkel
7	bank
7	u
6	shares
6	deutsche
6	via
5	issue
5	slide
5	perception
5	last
5	admits
4	hillary
4	de

Save Image

Figure 2.13. Orange Data Mining Tool (working with Twitter platform).
Source: https://bit.ly/bigdata2020-orange

We encourage you to explore the possibilities not only of that one chosen data science tool, but also other tools presented in Table 2.2.

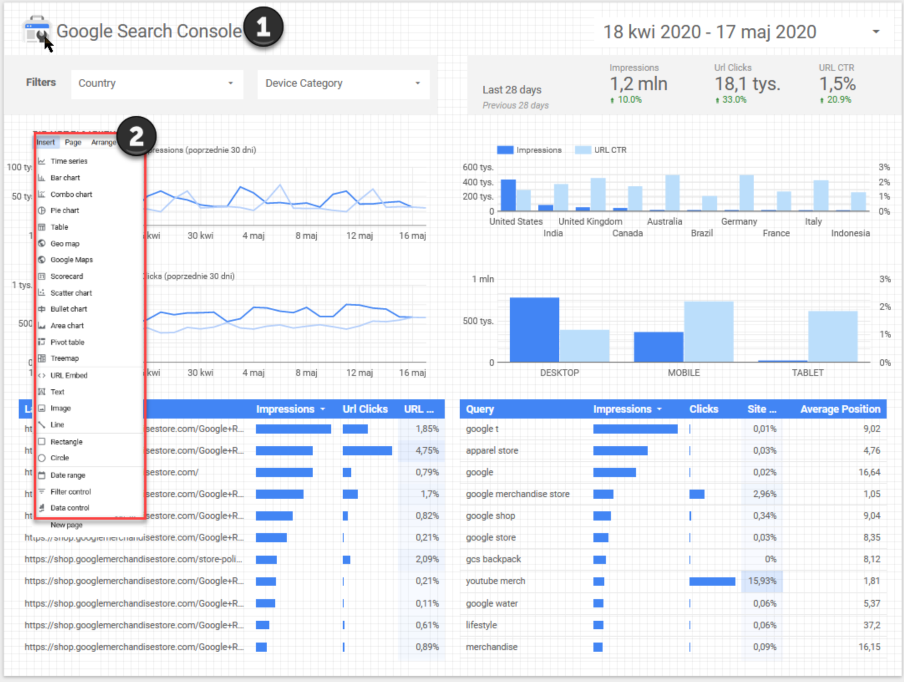
In the next section, we will show three examples showing data science tools in action.

2.5. Examples of use

In this subsection, we will show two short demonstrations of chosen data science tools: Google data studio and Flourish.

Let us start with Google data studio. In Figure 2.14 we can see a screenshot presenting the dashboard that has been created based on sample data, as well as (2) the list of objects that we can insert into the dashboard.

It is just a simple example using sample data sources (*small data*). As we assume, when we switch from *small data* into *big data*, it will be still important to build the output (preferably looking like the dashboard) with some interactivity (e.g., ability to apply filters, reconfigure the dashboard (decide what elements shall be visible and where they shall be placed), modify the theme, customize the particular part of the dashboard, etc.).



L	Impressions	Url Clicks	URL ...
ht sestore.com/Google+R...			1,85%
ht sestore.com/Google+R...			4,75%
ht sestore.com/			0,79%
ht sestore.com/Google+R...			1,7%
ht sestore.com/Google+R...			0,82%
https://shop.googlemerchandisestore.com/Google+R...			0,21%
https://shop.googlemerchandisestore.com/store-poli...			2,09%
https://shop.googlemerchandisestore.com/Google+R...			0,21%
https://shop.googlemerchandisestore.com/Google+R...			0,11%
https://shop.googlemerchandisestore.com/Google+R...			0,61%
https://shop.googlemerchandisestore.com/Google+R...			0,89%

Query	Impressions	Clicks	Site ...	Average Position
google t			0,01%	9,02
apparel store			0,03%	4,76
google			0,02%	16,64
google merchandise store			2,96%	1,05
google shop			0,34%	9,04
google store			0,03%	8,35
gcs backpack			0%	8,12
youtube merch			15,93%	1,81
google water			0,06%	5,37
lifestyle			0,06%	37,2
merchandise			0,09%	16,15

Figure 2.14. Google data studio (sample report). *Source:* https://datastudio.google.com

We start with a data source, next we create the dashboard and finally, we use the created dashboard. We can move further, for example by sharing the dashboard and using it in a particular context (e.g., a project). There is a possibility of using Google data studio for big data projects.

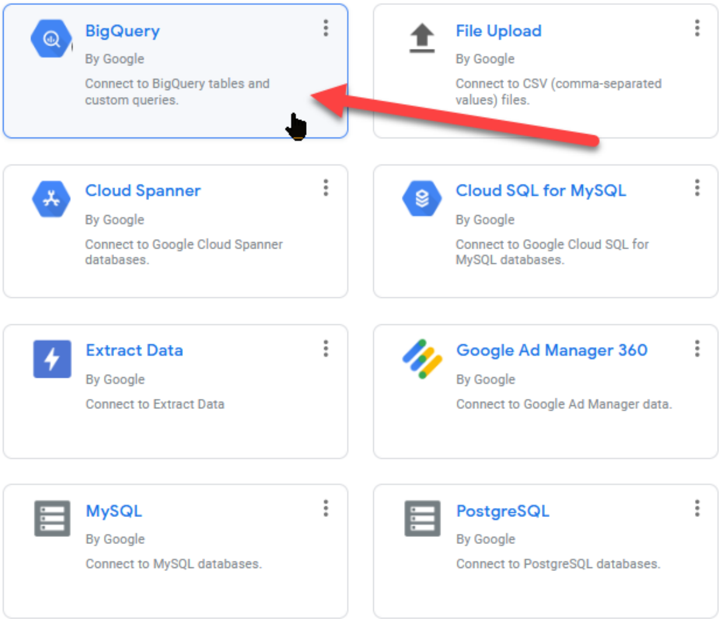
At the beginning, instead of creating the report, we start with creatinga data *source*.

For the time being, there are 17 Google connectors, 189 partner connectors and 3 open-source connectors.

Let us take a look at some examples of provided partner and open-source connectors:

- eBay;
- Facebook Ads;
- Google Analytics;
- Kaggle;
- LinkedIn Ads;
- Microsoft Ads;
- PayPal;
- Reddit.

And now we explore the connectors built and supported by Google.



BigQuery
By Google
Connect to BigQuery tables and custom queries.

File Upload
By Google
Connect to CSV (comma-separated values) files.

Cloud Spanner
By Google
Connect to Google Cloud Spanner databases.

Cloud SQL for MySQL
By Google
Connect to Google Cloud SQL for MySQL databases.

Extract Data
By Google
Connect to Extract Data

Google Ad Manager 360
By Google
Connect to Google Ad Manager data.

MySQL
By Google
Connect to MySQL databases.

PostgreSQL
By Google
Connect to PostgreSQL databases.

Figure 2.15. Google data studio connectors. *Source:* https://datastudio.google.com

As we can see in Figure 2.15, it is possible to use Google data studio with big data, through BigQuery. In section 2.7 we will show an example of Google BigQuery.

The second example is based on a software tool called *Flourish Studio*, which is available at: https://app.flourish.studio.

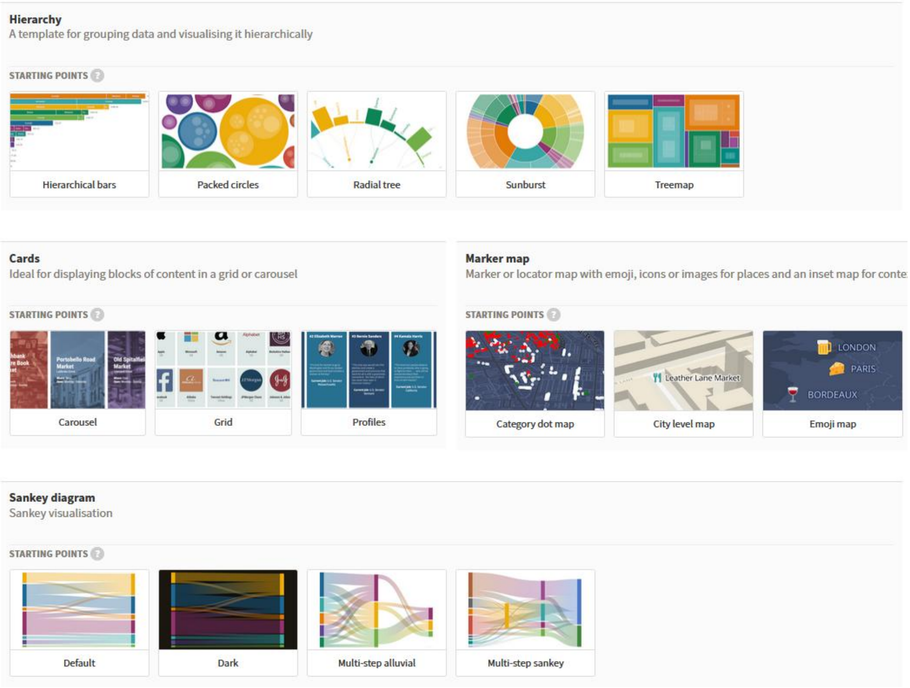
We will use a public account, it will be sufficient from the point of view of our requirements.

After successful signing up, we can choose one from predefined templates, which are categorized into:

a) line, bar and pie charts,
b) projection map,
c) scatter,
d) hierarchy,
e) bar chart race,
f) cards,
g) marker map,
h) sankey diagram,
i) connections globe,
j) icon map,
k) line chart race,
l) photo slider,
m) table,

n) chord diagram,
o) network graph,
p) parliament chart,
q) and more.

The Flourish Studio welcome screen is shown in Figure 2.16.



Hierarchy
A template for grouping data and visualising it hierarchically

Cards
Ideal for displaying blocks of content in a grid or carousel

Marker map
Marker or locator map with emoji, icons or images for places and an inset map for conte

Sankey diagram
Sankey visualisation

Figure 2.16. Flourish studio. *Source: Author's own*

Flourish Studio is just one of many examples of data science software that can be used in our projects. Due to its functionality, it will play a supportive role in creating big data solutions.

For example: Google data studio (or any similar alternative software) can be used for performing *search tasks* in large datasets, while Flourish Studio (or any similar alternative software) can be used for *demonstration purposes*.

More examples can be found in [2.5], [2.7] or [2.14].

2.6. Introduction to Big Data Ecosystem

We assume for the purpose of this book that we will use open source big data solutions from Apache Software Foundation, precisely *Apache Hadoop* with a collection of additional software packages.

In this subsection we will just enumerate the base elements and we will describe Apache Hadoop in more detail in chapter 3.

Apache framework consists of:

- Hadoop Common – libraries and utilities needed by Hadoop modules;
- Hadoop Distributed File System (HDFS) – a distributed file system that stores data;
- Hadoop YARN – a platform responsible for managing computing resources;
- Hadoop MapReduce – a programming model for large-scale data processing.

The list of additional software packages is extensive, we will mention just a few of them:

- Apache Pig is a language used to run jobs on Hadoop.;
- Apache Hive provides querying capabilities to view data;
- Apache HBase is an open-source, non-relational, distributed database;
- Apache Phoenix is an open-source, massively parallel, relational database engine;
- Apache Spark is a distributed cluster-computing framework.

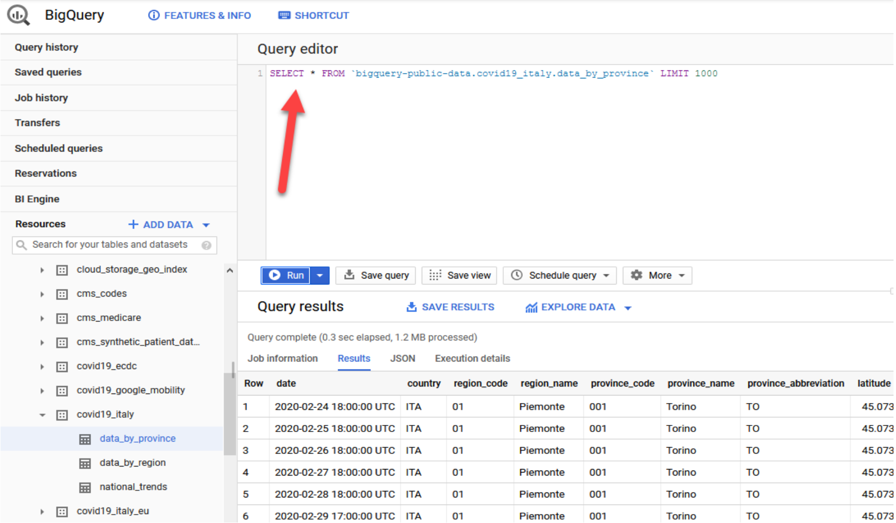
There are many scientific publications describing the big data ecosystem, for example: [2.1], [2.2], [2.3], [2.6], [2.8], [2.9], [2.10], [2.11], [2.12], [2.13], [2.15], [2.16] and [2.17].

In Appendix A.2, a big data landscape has been shown (in graphic and table form).

2.7. AN EXAMPLE OF SELECTED BIG DATA SOLUTION

For the purpose of this subsection, we will provide one short example of big data solution. Let us take a look at Google Big Query.

On Figure 2.17 we can see a screenshot of Google Big Query console with the query editor, the query, and query results. The query was run against data related to COVID-19 in Italy.



```
SELECT * FROM `bigquery-public-data.covid19_italy.data_by_province` LIMIT 1000
```

Query complete (0.3 sec elapsed, 1.2 MB processed)

Row	date	country	region_code	region_name	province_code	province_name	province_abbreviation	latitude
1	2020-02-24 18:00:00 UTC	ITA	01	Piemonte	001	Torino	TO	45.073
2	2020-02-25 18:00:00 UTC	ITA	01	Piemonte	001	Torino	TO	45.073
3	2020-02-26 18:00:00 UTC	ITA	01	Piemonte	001	Torino	TO	45.073
4	2020-02-27 18:00:00 UTC	ITA	01	Piemonte	001	Torino	TO	45.073
5	2020-02-28 18:00:00 UTC	ITA	01	Piemonte	001	Torino	TO	45.073
6	2020-02-29 17:00:00 UTC	ITA	01	Piemonte	001	Torino	TO	45.073

Figure 2.17. Google BigQuery. *Source: Author’s own.*

2.8. Further work

As we are approaching the end of the second chapter, it will be good to show the list of chosen data science challenges (see Tab. 2.3).

Table 2.3. Data science challenges

ID	Name	URL
DC1	Crowdanalytix	https://www.crowdanalytix.com/community
DC2	NumerAI	https://docs.numer.ai/tournament/learn
DC3	CrowdAI	https://www.crowdai.org/
DC4	DrivenData	https://www.drivendata.org/
DC5	ML Contest	https://mlcontests.com/
DC6	Zindi	https://zindi.africa/about
DC7	Smogathon	https://smogathon.com/pl/strona-glowna-old/
DC8	Codalab	https://competitions.codalab.org/
DC9	TunedIT	http://tunedit.org/challenges
DC10	Kaggle	https://www.kaggle.com/competitions

Source: Author's own

We believe this can be a good idea, to apply the method of getting more knowledge on-challenge basis. Best of luck!

2.9. Summary

In the second chapter, we have elaborated the big data ecosystem.

From opening examples related with data visualization, through data sources and data tools, finally describing the big data landscape.

Please go through the references itemized below and explore in more detail the chosen topics which you are interested in.

In the next, third chapter, we will cover the topics related with practical aspects of working with big data.

2.10. EXERCISES

1. Characterize data visualization example chosen from the Internet references section.
2. Explore the project "Word tree" (see: https://www.jasondavies.com/wordtree/).
3. Define an art diagram and prepare an example illustrating its usage.
4. Prepare a word cloud visualization based on keyword: "big data".
5. Pick a data source from the tab. X and explore it.
6. Pick two data science tools from the tab. Y and prepare a comparative analysis.
7. What is Google MapReduce?
8. What is Apache Hadoop?
9. What is Apache Spark?
10. Provide an example of a big data tool available as software as a service.
11. Analyze proposed list challenges. Which one would you choose and why?

2.11. References

2.11.1. Bibliography

[2.1] Akhtar, Aleem, *Role of Apache Software Foundation in Big Data Projects*, arXiv:2005.02829 [cs], May 2020, arXiv.org, http://arxiv.org/abs/2005.02829.

[2.2] Albert Y. Zomaya, Sherif Sakr (2017), *Handbook of Big Data Technologies*, Springer.

[2.3] Bautista E., Whitney C., Davis T. (2016), *Big Data Behind Big Data*, In: Arora R. (eds.), *Conquering Big Data with High Performance Computing*, Springer, Cham.

[2.4] Chaki S. (2015), *Pillar No. 8: Big Data Components*, In: *Enterprise Information Management in Practice*, Apress, Berkeley, CA.

[2.5] Divya Zion G., Tripathy B.K. (2020), *Comparative Analysis of Tools for Big Data Visualization and Challenges*, In: Anouncia S., Gohel H., Vairamuthu S. (eds.), *Data Visualization*, Springer, Singapore.

[2.6] Elgendy N., Elragal A. (2014), *Big Data Analytics: A Literature Review Paper*, In: Perner P. (eds.), *Advances in Data Mining. Applications and Theoretical Aspects*, ICDM 2014. Lecture Notes in Computer Science, vol 8557, Springer, Cham.

[2.7] Furht B., Villanustre F. (2016), *Introduction to Big Data*, In: *Big Data Technologies and Applications*, Springer, Cham.

[2.8] Luengo J., García-Gil D., Ramírez-Gallego S., García S., Herrera F. (2020), *Big Data Software*, In: *Big Data Preprocessing*, Springer, Cham.

[2.9] Luengo J., García-Gil D., Ramírez-Gallego S., García S., Herrera F. (2020), *Big Data: Technologies and Tools*, In: *Big Data Preprocessing*, Springer, Cham.

[2.10] Mazumder S. (2016), *Big Data Tools and Platforms*, In: Yu S., Guo S. (eds.), *Big Data Concepts, Theories, and Applications*, Springer, Cham.

[2.11] Odegua, Rising, Festus Ikpotokin, *DataSist: A Python-based library for easy data analysis, visualization and modeling*, arXiv:1911.03655 [cs], styczeń 2020, arXiv.org, http://arxiv.org/abs/1911.03655.

[2.12] Prabhu C., Chivukula A., Mogadala A., Ghosh R., Livingston L. (2019), *Big Data Analytics for Insurance*, In: *Big Data Analytics: Systems, Algorithms, Applications*, Springer, Singapore.

[2.13] Rehman, Arshia et al., *Leveraging Big Data Analytics in Healthcare Enhancement: Trends, Challenges and Opportunities*, arXiv:2004.09010 [cs, stat], kwiecień 2020, arXiv.org, http://arxiv.org/abs/2004.09010.

[2.14] Rodríguez-Molano J.I., Contreras-Bravo L.E., López-Santana E.R. (2018), *Big Data Tools for Smart Cities*, In: Tan Y., Shi Y., Tang Q. (eds.), *Data Mining and Big Data*, DMBD 2018. Lecture Notes in Computer Science, vol 10943, Springer, Cham.

[2.15] Sawant N., Shah H. (2013), *Big Data Deployment Patterns*, In: *Big Data Application Architecture Q & A*, Apress, Berkeley, CA.

[2.16] Wadkar S., Siddalingaiah M. (2014), *Motivation for Big Data*, In: *Pro Apache Hadoop*, Apress, Berkeley, CA.

[2.17] Yoo S., Choi K., Lee M. (2014), *Business Ecosystem and Ecosystem of Big Data*, In: Chen Y. et al. (eds.), *Web-Age Information Management*, WAIM 2014. Lecture Notes in Computer Science, vol 8597, Springer, Cham.

[2.18] Yoshida N. (2015), *A Large Sky Survey Project and the Related Big Data Analysis*, In: Chu W., Kikuchi S., Bhalla S. (eds.), *Databases in Networked Information Systems*, DNIS 2015. Lecture Notes in Computer Science, vol 8999, Springer, Cham.

2.11.2. INTERNET REFERENCES

[2.19] https://www.babynamewizard.com/voyager

[2.20] https://projects.fivethirtyeight.com/complete-history-of-the-nfl

[2.21] https://podio.com/site/creative-routines

[2.22] http://www.slate.com/blogs/the_slatest/2015/10/06/syrian_conflict_relationships_explained.html

[2.23] https://www.informationisbeautiful.net/visualizations/the-hollywood-insider

[2.24] https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks

[2.25] http://hint.fm/wind

[2.26] https://flowingdata.com/2015/12/15/a-day-in-the-life-of-americans

[2.27] https://www.cloudred.com/labprojects/nyctrees

[2.28] https://www.nikon.com/about/sp/universcale/scale.htm

[2.29] https://www.washingtonpost.com/graphics/2019/business/immersive-space-suits-history-fashion-and-function

[2.30] https://www.nationalgeographic.com/science/2019/07/the-atlas-of-moons

[2.31] https://pudding.cool/2018/08/wiki-death

[2.32] https://www.jasondavies.com/wordtree

[2.33] https://textvis.lnu.se

[2.34] https://treevis.net

[2.35] https://www.c82.net/work/?id=347

[2.36] http://vcg.informatik.uni-rostock.de/~ct/timeviz/timeviz.html

[2.37] https://sentimentvis.lnu.se

[2.38] https://www.c82.net/euclid

[2.39] http://graphics.reuters.com/TECHNOLOGY-BLOCKCHAIN/010070P11GN/index.html

[2.40] https://symbolikon.com/all-symbols-general-gallery

[2.41] https://www.nytimes.com/interactive/2015/05/17/us/elections/2016-presidential-campaigns-staff-connections-clinton-bush-cruz-paul-rubio-walker.html

[2.42] https://www.bloomberg.com/graphics/2015-whats-warming-the-world

[2.43] http://graphics.wsj.com/infectious-diseases-and-vaccines

[2.44] https://qz.com/296941/interactive-graphic-every-active-satellite-orbiting-earth

[2.45] https://money.cnn.com/interactive/economy/diversity-millennials-boomers

[2.46] https://earth.nullschool.net

[2.47] https://www.pinterest.it/pin/137993176054150421

[2.48] https://fathom.info/traces

[2.49] http://www.puffpuffproject.com/languages.html

[2.50] https://projects.fivethirtyeight.com

[2.51] http://vis.stanford.edu/files/2007-AnimatedTransitions-InfoVis.pdf

Chapter 3. Working with big data

Torture the data, and it will confess to anything.

R. Coase

3.1. INTRODUCTION

The third (and last) chapter is the most practical part of this handbook. There are a lot of literature items that are related to this topic, for example: [3.1], [3.2] and [3.3].

This chapter has been divided into two main sections. The first section consists of a description of the chosen big data ecosystem, namely Apache Hadoop. The second section is devoted to a presentation of practical examples of use of chosen big data solution (hands-on-lab approach).

While the examples only require basic Python language knowledge, basic machine learning knowledge is appreciated and will help in understanding the generic workflow.

3.2. APACHE HADOOP

Apache Hadoop is a collection of software used for distributed file storage and manipulation of data stored in it. A default download package contains Hadoop Distributed File System (HDFS), Hadoop YARN, and Hadoop MapReduce.

HDFS is a central part of the whole ecosystem, allowing for storing data in a distributed manner on multiple computers. Singular name nodes, also called master nodes, are responsible for tracking file names and their location on data nodes, as well as data replication and management of data nodes. Each one of many data nodes stores parts of files as blocks. Data is stored redundantly, so in case of data node failure, the name node can detect it and redirect all queries related to files stored on the failed data node to other nodes containing copies of needed resources. Apache Hadoop offers multiple options of abstraction, so the end-user can easily integrate this distributed data storage into other systems. These options include native java interface, HTTP REST API, or NFS share. Other Apache projects can also build upon data stored in HDFS, for instance, Apache Hive allows for SQL operations on stored data, and contains JDBC driver for interoperability.

Apache YARN consists of two separate services responsible for managing resources of the cluster and tasks execution. The singular resource manager receives information about resource usage of each machine, as well information about each running task progress from each application master, and assigns tasks to computers accordingly. Its scheduler can be changed depending on whether the user needs uniform resources allocation or allocation based on job importance. The resource manager is also responsible for maintaining a list of jobs, taking in new ones and restating ones that failed. A reservation system is also available for time-constrained tasks, so they can be executed within predictable time. Each computer runs a node manager service that reads its RAM, CPU and disk usages and reports them to the resource manager.

Apache MapReduce parallelizes calculations that can be done on each row of data independently by sending instructions to each computer, which in typical configuration contains both compute node and data node, for it to execute them on part of a dataset stored locally on a machine. This part is called mapping, and after all operations are finished, the reduce operation is performed on all data, which gathers partial data from each node, for instance a reducer can count or sum data stored in each column, effectively combining data from all rows into one final result. There is a growing popularity of Apache Spark, which is also a distributed computation framework, which, in certain circumstances, can be up to 100 times faster than Apache MapReduce.

Apache Spark is a framework that allows us to do computations on local and distributed data. While this framework was written in Scala, there are official binds for Scala, Java, Python and R. While many interfaces are available in all languages, some of them are only available in one language, especially newer classes tend to be available in Scala and Java first before they arrive to other languages.

Originally, data processed by Spark library was stored in resilient distributed dataset (RDD), but usage of `Dataframe` class for storing data is more advised because Apache Spark machine learning library for data stored in RDD manner was deprecated in version 2.0 and its support is expected to be removed in version 3.0 of Apache Spark framework. Since version 2.0 the data in Apache Spark can be represented by the `DataFrame` class. Data can be read from many formats, like CSV, JSON, Parquet or from distributed data storage, for instance HDFS. Apache Spark also contains JDBC driver, making it possible for it to act as a distributed query engine.

Operations on `Dataframes` include selecting rows and columns, filtering them, renaming columns, as well as adding new columns filled with predefined data or calculated formulas (Fig. 3.1). DataFrame objects also support the execution of SQL commands, making them easy to use for anyone with database management experience. Apache Spark provides a variety of functions that can transform DataFrame, like column formatting, filling or removing rows with empty cells or grouping data and counting.

Apache Spark also contains Spark Streaming library, which can read live data from streams, and process it in batches. Data can be read from many sources, including TCP sockets, Kafka, or file streams.

Another component of Apache Spark is its machine learning library, called **MLlib**.

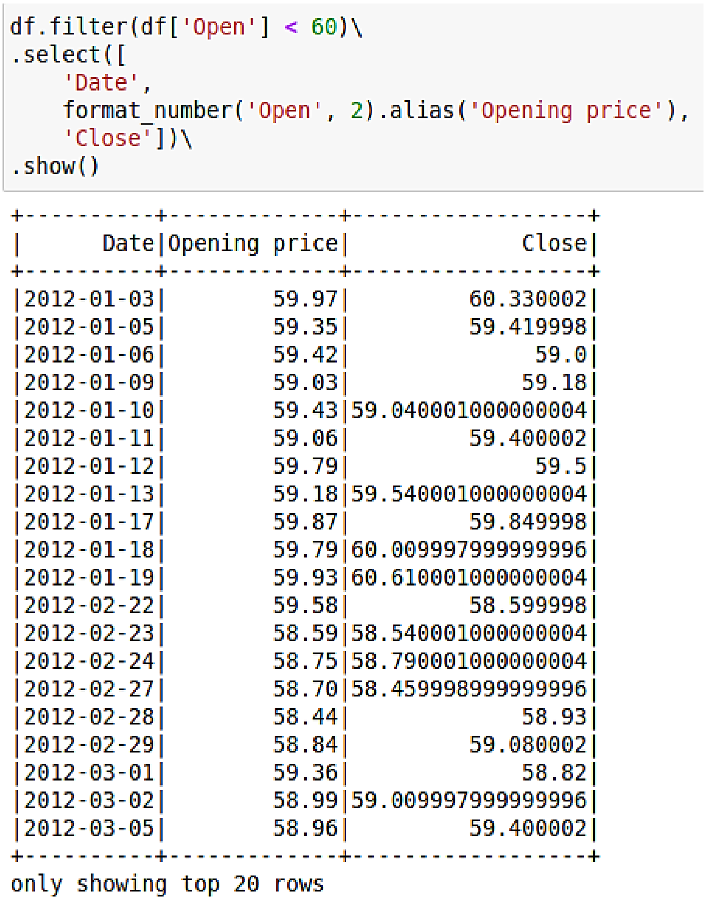
It contains many interfaces related to regression, clustering and classification, containing tools for both supervised and unsupervised machine learning as well for evaluation of created models. Some of the included algorithms are linear and logistic regressions, decision trees Gradient-Boosted Trees, and K-means clustering. Many classes have similar construction making it easy to substitute them or to create automated pipelines. For most algorithms, we first define a new object containing parameters of the algorithm used, for instance names of features and label columns for linear regression. Then, we invoke `fit(DataFrame)` method on a previously defined model definition, with training data passed to it as a parameter, which creates a new model object that can transform data. The trained model contains a `transform(DataFrame)` method that transforms data, usually by adding new columns with calculated data. MLlib provides various evaluators that can be used on a dataset transformed by a model to get metrics on how well a model behaves, for instance mean squared error for regression models or silhouette for clustering models.

Many operations can be chained together with a help of `Pipeline` object, which takes in a list of defined objects that have `fit()` or `evaluate()` methods and can be used on `DataFrame` either by invoking `fit()` or `evaluate()` method on Pipeline object itself, in turn passing data to respective methods of each object in a list, allowing us to easily chain multiple steps into just one.

Typical workflow for machine learning starts with declaring a Spark session, which holds configuration and application-specific data. Then the data we want to work on is loaded, either from local storage, distributed storage system, or is streamed from a live source. The data schema can either be automatically deducted from data itself or provided by the user in a form of row name, type of stored data and optional flag if cells in the column can be nullable. After the data is loaded it can be prepared for analysis. These preparations can include, for instance, formatting data in columns, changing string labels into numerical indexes, splitting data from one column into multiple or dropping rows with empty cells.

After the dataset is prepared, we split it into training and test datasets used later for training machine learning models and testing its accuracy. In the next step, we declare which machine learning method we want to use and its options and train a new model on the training dataset from the previous step.

After the model is created we can use it on the test data to get estimates, which can be fed to one of included in Spark evaluators. If trained model characteristics meet our expectations we can then, for example, use it on other data to get predictions or save the model to file for future use (Fig. 3.1).



```
df.filter(df['Open'] < 60)\
.select([
    'Date',
    format_number('Open', 2).alias('Opening price'),
    'Close'])\
.show()
```

```
+----------+-------------+------------------+
|      Date|Opening price|             Close|
+----------+-------------+------------------+
|2012-01-03|        59.97|         60.330002|
|2012-01-05|        59.35|         59.419998|
|2012-01-06|        59.42|              59.0|
|2012-01-09|        59.03|             59.18|
|2012-01-10|        59.43|59.040001000000004|
|2012-01-11|        59.06|         59.400002|
|2012-01-12|        59.79|              59.5|
|2012-01-13|        59.18|59.540001000000004|
|2012-01-17|        59.87|         59.849998|
|2012-01-18|        59.79|60.009997999999996|
|2012-01-19|        59.93|60.610001000000004|
|2012-02-22|        59.58|         58.599998|
|2012-02-23|        58.59|58.540001000000004|
|2012-02-24|        58.75|58.790001000000004|
|2012-02-27|        58.70|58.459998999999996|
|2012-02-28|        58.44|             58.93|
|2012-02-29|        58.84|         59.080002|
|2012-03-01|        59.36|             58.82|
|2012-03-02|        58.99|59.009997999999996|
|2012-03-05|        58.96|         59.400002|
+----------+-------------+------------------+
only showing top 20 rows
```

Figure 3.1. Example of filtering data by Open column and displaying formatted data from selected columns. *Source: Author's own*

3.3. PYSPARK HANDS-ON-LAB

In this section, we will focus on the installation and configuration of pySpark and Apache Hadoop for its HDFS component on Linux Ubuntu. We assume that Java Runtime Environment 8, Python in version at least 3.4 and pip package manager are installed.

We need to install a Python component that allows interfacing with Java named `py4j`. We also need to install Scala language libraries used by Apache Spark (Listing 3.1). Optionally we can also install Jupyter notebook for ease of use. It can be later invoked with the "`jupyter notebook`" command. Apache Hadoop requires `ssh` server and `pdsh` utility.

```
sudo apt install scala

pip3 install py4j

pip3 install jupyter

sudo apt install ssh

sudo apt install pdsh
```

Listing 3.1. Installation of required dependencies, *Source: Author's own*

The computer needs to be able to connect to itself through passphraseless ssh connection for Apache Hadoop to work. We can test this and we can issue the following command (Listing 3.2).

```
ssh localhost
```

Listing 3.2. Installation of required dependencies, *Source: Author's own*

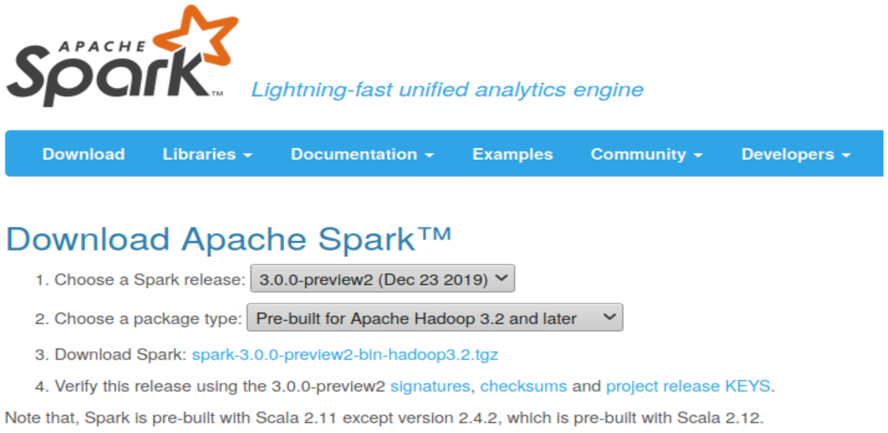
If this command fails we need to create a new ssh key and add it to the list of authorized keys. Ssh may not allow for connection if the permissions of that list files are too broad, so we only allow users to read and write that file (Listing 3.3).

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

chmod 0600 ~/.ssh/authorized_keys
```

Listing 3.3. Installation of required dependencies, *Source: https://bit.ly/bigdata-2020-hadoop*

We go to `https://hadoop.apache.org/releases.html`, to download and unpack the 3.2.1 version.

Then we go to `https://spark.apache.org/downloads.html`. We will be using version 3.0.0-preview2, pre-built for Apache Hadoop 3.2 and later. We download and unpack the archive where it will be used (Fig. 3.2).



Download Apache Spark™

1. Choose a Spark release: 3.0.0-preview2 (Dec 23 2019)
2. Choose a package type: Pre-built for Apache Hadoop 3.2 and later
3. Download Spark: spark-3.0.0-preview2-bin-hadoop3.2.tgz
4. Verify this release using the 3.0.0-preview2 signatures, checksums and project release KEYS.

Note that, Spark is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12.

Figure 3.2. Apache Spark download page with selected release used in our examples.
Source: Author's own

To set up Apache Hadoop we first edit `etc/hadoop/hadoop-env.sh` file contained in unpacked directory to contain path to Java virtual machine (Listing 3.4), in our case we needed to set `JAVA_HOME` variable to path of our installation of Java Virtual Machine, as well as adding the system variable with the same name (Listing 3.5).

```
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Listing 3.4. Changed part of etc/hadoop/hadoop-env.xml file, *Source: Author's own*

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Listing 3.5. Setting system variable in terminal, *Source: Author's own*

After this, we set Hadoop to operate in a single-node pseudo-distributed node for simplicity. We set a default path to simplify command line interface usage to omit part of URL during operation on files (Listing 3.6) and change replication value to one since we will only use one node(Listing 3.7).

```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:8020</value>
    </property>
</configuration>
```

Listing 3.6. Changed part of etc/hadoop/core-site.xml file. *Source: Author's own*

```
<configuration>
  <property>
        <name>dfs.replication</name>
        <value>1</value>
  </property>
</configuration>
```

Listing 3.7. Changed part of etc/hadoop/hdfs-site.xml file. *Source: Author's own*

To use Apache Spark we set the environment variable "`SPARK_HOME`" to the directory where Spark was unpacked to. Then we set the environment variable "`PYSPARK_PYTHON`" to the installed Python 3 executable.

Thereafter, we add the path to Spark python directory to `PYTHONPATH` variable.

We also set `PYSPARK_DRIVER_PYTHON` to the preferred execution environment, for instance python3 or Jupyter notebook. In our case, we want to use Jupyter notebook, so we have set the default interpreter to Jupyter by defining "`PYSPARK_DRIVER_PYTHON`" and its options "`PYSPARK_DRIVER_PYTHON_OPTS`" to the notebook. We can check if installation was successful by writing "`import pyspark`" in Python (Listing 3.8). If the import is completed successfully then Spark and pySpark are ready to use.

```
export SPARK_HOME='/home/user/spark-3.0.0-preview2-bin-
hadoop3.2'

export PYTHONPATH=$SPARK_HOME/python:$PYTHONPATH

export PYSPARK_PYTHON=python3

export PYSPARK_DRIVER_PYTHON="jupyter"

export PYSPARK_DRIVER_PYTHON_OPTS="notebook"
```

Listing 3.8. System variables needed for pySpark. *Source: Author's own*

In the first example we will use public Iris dataset [3.4] and try to determine a flower class with the help of logistic regression, which assigns a numeric class for each dataset row. First, we need to import the SparkSession class and initialize the new session by declaring a new SparkSession object and providing its name (Listing 3.9).

```
from pyspark.sql import SparkSession

spark = SparkSession.Builder().appName('iris').getOrCreate()
```

Listing 3.9. Initializing Spark session. *Source: Author's own*

When loading data, Spark can either try to automatically infer schema from data or we can provide schema. The former is slower because Spark needs to read data twice, first time to infer schema, and second to read data itself. Providing a schema requires us to write a structure with all column names, their types, and optional information if the column is nullable (Listing 3.10).

```
from pyspark.sql.types import StructType, StructField,
DecimalType, StringType

data_schema = StructType([

    StructField('sepal_l', DecimalType(scale=1), False),

    StructField('sepal_w', DecimalType(scale=1), False),

    StructField('petal_l', DecimalType(scale=1), False),

    StructField('petal_w', DecimalType(scale=1), False),

    StructField('class', StringType(), False)

])
```

Listing 3.10. Defining dataset schema. *Source: Author's own*

When we want to load CSV data we need to provide file location, its schema and if the first row of data contains columns headers. If we wanted to infer schema, we would use `inferSchema` parameter instead of schema, however, this option is slower because it needs to read data twice (Listing 3.11).

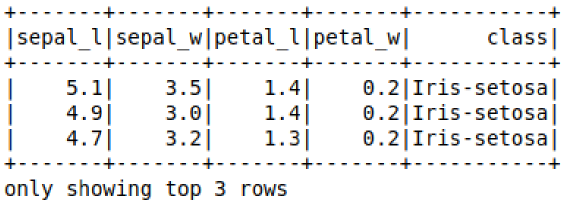
```
df = spark.read.csv(

    'iris.data', header=False,

    schema=data_schema)

# df = spark.read.csv(

    'iris.data', header=False,

    inferSchema=True)
```

Listing 3.11. Loading data from local csv file. *Source: Author's own*

We can check data by calling `show()` method of `DataFrame` object, with optional parameters containing number of rows to display (Listing 3.12) (Fig. 3.3). By default, Spark shows the first 20 rows of data.

```
df.show(3)
```

Listing 3.12. Showing first three rows of DataFrame. *Source: Author's own*



```
+-------+-------+-------+-------+-----------+
|sepal_l|sepal_w|petal_l|petal_w|      class|
+-------+-------+-------+-------+-----------+
|    5.1|    3.5|    1.4|    0.2|Iris-setosa|
|    4.9|    3.0|    1.4|    0.2|Iris-setosa|
|    4.7|    3.2|    1.3|    0.2|Iris-setosa|
+-------+-------+-------+-------+-----------+
only showing top 3 rows
```

Figure 3.3. First three rows of DataFrame containing data from iris dataset. *Source: Author's own*

After data is loaded, we can start preparing data for a logistic regression algorithm. First, we map a numerical index to each flower class, as they are stored as text. `StringIndexer` requires us to define the name of the input textual column and new output column with indexes (Listing 3.13). Note that this operation only defines how data should be changed but does not change it in this step. If we wanted to add "class_index" column to our data then we would need to invoke indexer.fit(DataFrame) method to create StringIndexerModel which stores information on class name to its index transformation.

```
from pyspark.ml.feature import StringIndexer

indexer = StringIndexer(

    inputCol='class', outputCol='class_index'

)
```

Listing 3.13. Creating StringIndexer object for class column. *Source: Author's own*

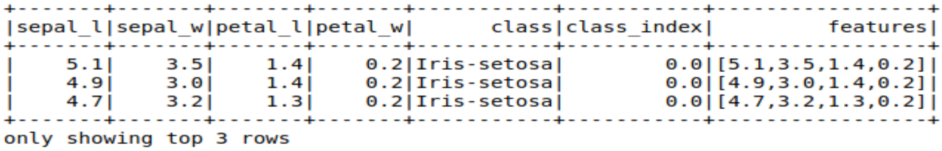
Invoking indexer.transform(DataFrame) on the created model would return a new DataFrame with added column.

Logistic regression algorithms require two columns, one with a vector of features used to determine labels, and a column containing numeric labels. We can combine columns into vectors with VectorAssembler class, which requires us to define a list of used columns names and name of output column (Listing 3.14). Usually the output column is named "features" because it is the default name of the features column used in regression model initialization.

```
from pyspark.ml.feature import VectorAssembler

features_assembler = VectorAssembler(

    inputCols=['sepal_l', 'sepal_w', 'petal_l', 'petal_w'],

    outputCol='features')
```

Listing 3.14. Creating VectorAssembler object. *Source: Author's own*

Returned features_Assembler object have transform(DataFrame) method, which, similarly to StringIndexer, returns new DataFrame object with features column added. We can see that our DataFrame now has two additional columns with prepared data (Fig. 3.4).



```
+-------+-------+-------+-------+-----------+-----------+-----------------+
|sepal_l|sepal_w|petal_l|petal_w|      class|class_index|         features|
+-------+-------+-------+-------+-----------+-----------+-----------------+
|    5.1|    3.5|    1.4|    0.2|Iris-setosa|        0.0|[5.1,3.5,1.4,0.2]|
|    4.9|    3.0|    1.4|    0.2|Iris-setosa|        0.0|[4.9,3.0,1.4,0.2]|
|    4.7|    3.2|    1.3|    0.2|Iris-setosa|        0.0|[4.7,3.2,1.3,0.2]|
+-------+-------+-------+-------+-----------+-----------+-----------------+
only showing top 3 rows
```

Figure 3.4. First three rows of DataFrame containing prepared data from iris dataset shown with final_data.show(3) method. *Source: Author's own*

LogisticRegression class by default uses columns named "features" and "labels", however, we can provide different column names, as well as additional parameters, such as number of iterations. (Listing 3.15)

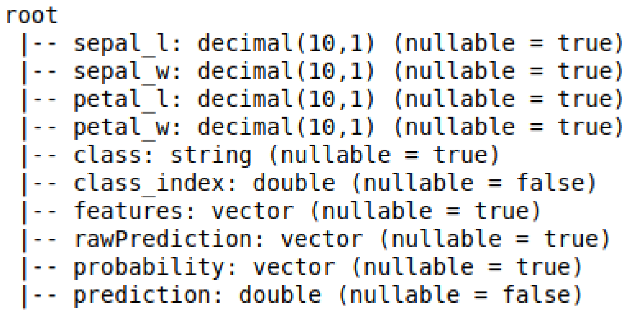
```
from pyspark.ml.classification import LogisticRegression
regressor =  LogisticRegression(
    featuresCol='features', labelCol='class_index'
)
```

Listing 3.15. Creating logisticRegression object. *Source: Author's own*

The regressor object also contains `fit(DataFrame)` method which returns a trained regression model. The trained model contains a `transform(DataFrame)` method, which returns `DataFrame` with added prediction columns (Fig. 3.6). We can invoke `printSchema()` method of `DataFrame` to show names and types of all of its columns (Listing 3.16).

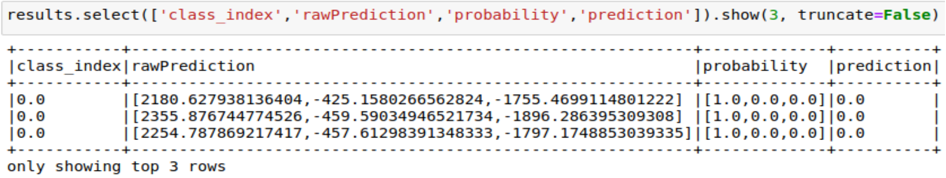
```
results.printSchema()
```

Listing 3.16. Printing schema of a DataFrame. *Source: Author's own*



```
root
 |-- sepal_l: decimal(10,1) (nullable = true)
 |-- sepal_w: decimal(10,1) (nullable = true)
 |-- petal_l: decimal(10,1) (nullable = true)
 |-- petal_w: decimal(10,1) (nullable = true)
 |-- class: string (nullable = true)
 |-- class_index: double (nullable = false)
 |-- features: vector (nullable = true)
 |-- rawPrediction: vector (nullable = true)
 |-- probability: vector (nullable = true)
 |-- prediction: double (nullable = false)
```

Figure 3.5. Schema of DataFrame containing training data with added prediction columns. *Source: Author's own*



```
results.select(['class_index','rawPrediction','probability','prediction']).show(3, truncate=False)
```

```
+-----------+-----------------------------------------------------------+-------------+----------+
|class_index|rawPrediction                                              |probability  |prediction|
+-----------+-----------------------------------------------------------+-------------+----------+
|0.0        |[2180.627938136404,-425.1580266562824,-1755.4699114801222] |[1.0,0.0,0.0]|0.0       |
|0.0        |[2355.876744774526,-459.59034946521734,-1896.286395309308] |[1.0,0.0,0.0]|0.0       |
|0.0        |[2254.787869217417,-457.61298391348333,-1797.1748853039335]|[1.0,0.0,0.0]|0.0       |
+-----------+-----------------------------------------------------------+-------------+----------+
only showing top 3 rows
```

Figure 3.6. Class index and added prediction columns for first three rows. *Source: Author's own*

Instead of invoking `fit()` and `transform()` methods on all previously declared objects, we can chain them together with the help of `Pipeline` class (Listing 3.17). The constructor takes in a list of objects and the returned object contains `fit(DataSet)` method, which invokes `fit()` and `transform()` methods on all objects that were passed during pipeline initialization. Each step takes in a DataFrame returned by the previous step, chaining them together. This function returns PipelineModel, which contains a `transform(DataFrame)` method that returns a new `DataFrame`.

```
from pyspark.ml import Pipeline
pipeline = Pipeline(stages=[indexer, features_assembler])
trained_pipeline = pipeline.fit(df)
final_data = trained_pipeline.transform(df)
```

Listing 3.17. Creating Pipeline object and creating new DataFrame with transformed data. *Source: Author's own*

We can split our prepared data into train and test datasets using `randomSplit` method, which takes in a list of percentages how data should be split (Listing 3.18). In our example we want 70% of all data rows to be split into train data, and 30% sent to test data.

```
train_data, test_data = final_data.randomSplit([0.7, 0.3])
```

Listing 3.18. Splitting prepared data into two separate DataFrame objects. *Source: Author's own*

We can now train our regression model on test data and get estimates for the test data (Listing 3.19).

```
regression_model = regressor.fit(train_data)
results = regression_model.transform(test_data)
```

Listing 3.19. Training logistic regression model and estimating classes for test data. *Source: Author's own*

Since we try to assign one of three classes to each row of data, we can use a multiclass evaluator object to determine how well we could determine class for each row. The evaluator takes in the name of the label column, name of predicted label, and a name of a metric we wish to obtain (Listing 3.20).

```
from pyspark.ml.evaluation import \
MulticlassClassificationEvaluator
evaluator = MulticlassClassificationEvaluator(
    predictionCol='prediction',
    labelCol='class_index', metricName='accuracy'
)
summary = evaluator.evaluate(results)
```

Listing 3.20. Creating MulticlassClassificationEvaluator object and calculating accuracy of our model. *Source: Author's own*

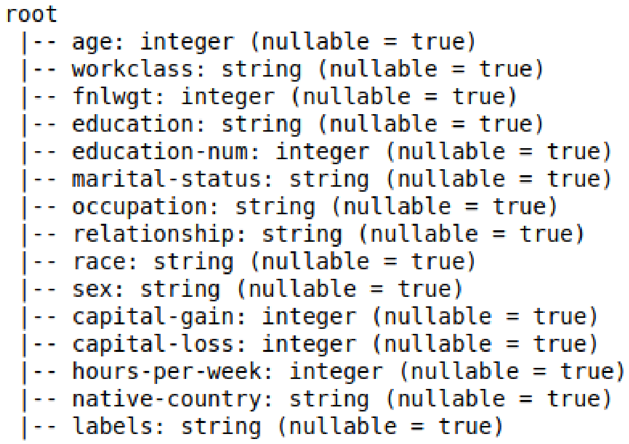
Returned summary is an accuracy of our model, in our case it was estimated at 0.9.

In the second example we will use the public adult income dataset [3.5], and try to determine if person income is less or more than $50000 yearly with help of decision trees. The decision tree on each step chooses one of two paths, depending on the value of a column until the lowest level of the tree is reached and a category is assigned to a row.

While we create SparkSession and data schema similarly as in the previous example, we modify the data delimiter in the next step because Spark by default assumes that CSV data is separated by comma, however, in this dataset cells are separated with comma and space (Listing 3.21) (Fig. 3.7).

```
df = spark.read.load('adult.data',

    format='csv', sep=', ',

    header=False, schema=data_schema

)
```

Listing 3.21. Loading data from local csv file with changed separator. *Source: Author's own*



```
root
 |-- age: integer (nullable = true)
 |-- workclass: string (nullable = true)
 |-- fnlwgt: integer (nullable = true)
 |-- education: string (nullable = true)
 |-- education-num: integer (nullable = true)
 |-- marital-status: string (nullable = true)
 |-- occupation: string (nullable = true)
 |-- relationship: string (nullable = true)
 |-- race: string (nullable = true)
 |-- sex: string (nullable = true)
 |-- capital-gain: integer (nullable = true)
 |-- capital-loss: integer (nullable = true)
 |-- hours-per-week: integer (nullable = true)
 |-- native-country: string (nullable = true)
 |-- labels: string (nullable = true)
```

Figure 3.7. Schema of loaded data. *Source: Author's own*

This dataset contains empty values, represented as "?". While DataFrame have built-in functions to deal with null values, we first need to replace all question marks with null values (Listing 3.22).

```
data = df.replace('?', None)
```

Listing 3.22. Replacing all cells containing question marks with null value, *Source: Author's own*

We can either fill empty cells with a value, for instance with mean value for numeric columns, or drop rows of data containing these empty cells (Listing 3.23). By default Spark drops rows where any cell is empty, but we can set the "how" parameter to "all" to remove only rows where all cells are empty.

```
data = data.na.drop(how='any')
```

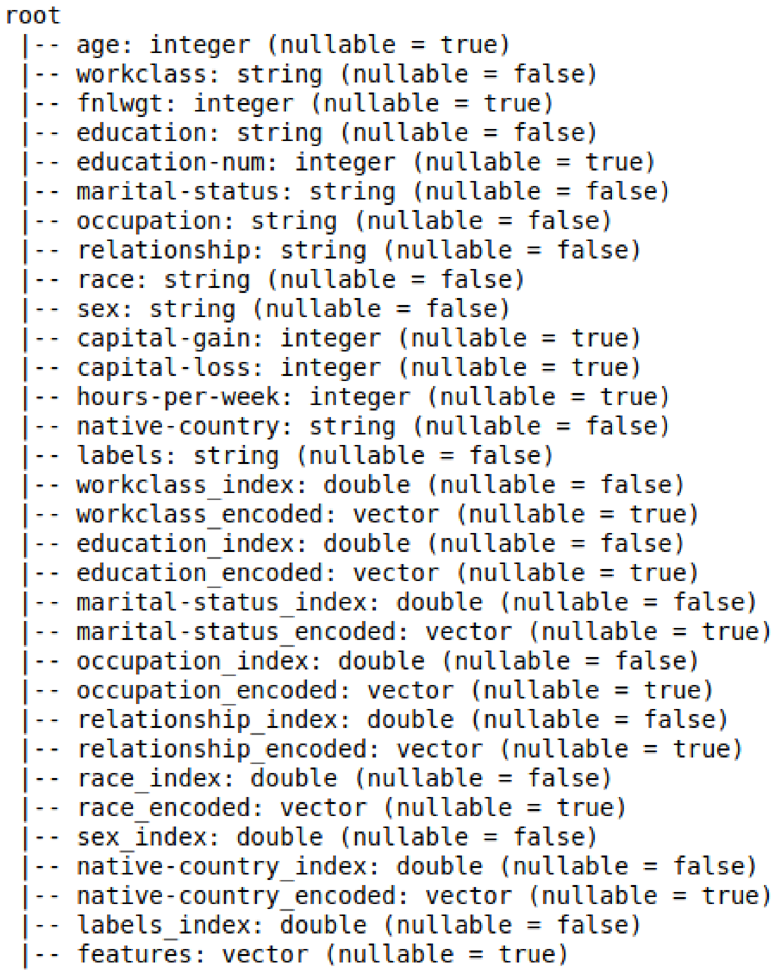
Listing 3.23. Removing all rows containing empty cells. *Source: Author's own*

We need to encode text columns, however, although in the first example we used `StringIndexer`, here we use it in combination with `HotOneEncoder` class (Listing 3.24). Where string indexer assigns a number for each class, the regression or classification model might wrongly assume that labels are ordered. One-hot creates as many binary columns as there are possible values, so that for each index value only one column is set.

```
workclass_indexer = StringIndexer(

   inputCol='workclass', outputCol='workclass_index'

)

workclass_encoder = OneHotEncoder(

    inputCol='workclass_index',
outputCol='workclass_encoded')
```

Listing 3.24. Encoding all text columns to separate columns for each possible column value.
Source: Author's own

After encoding all text columns we can create `VectorAssembler` and Pipeline objects similarly as in the previous example (Fig. 3.8). Note that pipeline adds new columns that are used to create labels_index and features columns, and the rest of the columns will not be used in later steps, so we can select just the columns we need for clarity.



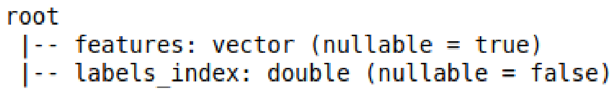
```
root
 |-- age: integer (nullable = true)
 |-- workclass: string (nullable = false)
 |-- fnlwgt: integer (nullable = true)
 |-- education: string (nullable = false)
 |-- education-num: integer (nullable = true)
 |-- marital-status: string (nullable = false)
 |-- occupation: string (nullable = false)
 |-- relationship: string (nullable = false)
 |-- race: string (nullable = false)
 |-- sex: string (nullable = false)
 |-- capital-gain: integer (nullable = true)
 |-- capital-loss: integer (nullable = true)
 |-- hours-per-week: integer (nullable = true)
 |-- native-country: string (nullable = false)
 |-- labels: string (nullable = false)
 |-- workclass_index: double (nullable = false)
 |-- workclass_encoded: vector (nullable = true)
 |-- education_index: double (nullable = false)
 |-- education_encoded: vector (nullable = true)
 |-- marital-status_index: double (nullable = false)
 |-- marital-status_encoded: vector (nullable = true)
 |-- occupation_index: double (nullable = false)
 |-- occupation_encoded: vector (nullable = true)
 |-- relationship_index: double (nullable = false)
 |-- relationship_encoded: vector (nullable = true)
 |-- race_index: double (nullable = false)
 |-- race_encoded: vector (nullable = true)
 |-- sex_index: double (nullable = false)
 |-- native-country_index: double (nullable = false)
 |-- native-country_encoded: vector (nullable = true)
 |-- labels_index: double (nullable = false)
 |-- features: vector (nullable = true)
```

Figure 3.8. Schema of prepared data. *Source: Author's own*

We have selected data from only labels_index and features columns, since these are the only two columns needed (Listing 3.25) (Fig. 3.9).

```
prepared_data = prepared_data.select(
    ['features', 'labels_index'])
prepared_data.printSchema()
```

Listing 3.25. Selecting and showing data schema from features and labels_index columns. *Source: Author's own*



```
root
 |-- features: vector (nullable = true)
 |-- labels_index: double (nullable = false)
```

Figure 3.9. Schema of prepared data with removed unnecessary columns. *Source: Author's own*

We can now declare our decision tree classifier. Spark provides `DecisionTreeClassifier`, `RandomForestClassifier`, and `GBTClassifier` classes. We will use a `RandomForestClassifier` class with just fifteen trees (Listing 3.26).

```
from pyspark.ml.classification import RandomForestClassifier

rf_classifier = RandomForestClassifier(

    featuresCol='features',

    labelCol='labels_index',

    numTrees=15

)
```

Listing 3.26. Creating RandomForestClassifier object with 15 random trees. *Source: Author's own*

Since we only assign data to one of two classes we can use `BinaryClassificationEvaluator` class instead of `MulticlassClassificationEvaluator` (Listing 3.27).

```
from pyspark.ml.evaluation import
BinaryClassificationEvaluator

bc_evaluator = BinaryClassificationEvaluator(

    rawPredictionCol='rawPrediction',

    labelCol='labels_index',

    metricName='areaUnderROC'

)

bc_results = bc_evaluator.evaluate(prediction)

print(bc_results)
```

Listing 3.27. Creating BinaryClassificationEvaluator object to check model accuracy. *Source: Author's own*

In our case, the area below the receiver operating characteristic curve is equal to about 0.99984.

In our third example, we will run Apache Hadoop in pseudo-distributed mode, upload files to a cluster and load it in Apache Spark. First, we format the filesystem stored on the cluster (Listing 3.28). We only need to do this once before the first start.

```
bin/hdfs namenode -format
```

Listing 3.28. Formatting filesystem used inside cluster. *Source: Author's own*

Then we can start name node and data node daemons. `Sbin` directory contains scripts responsible for starting and stopping various services provided by the Apache Hadoop package. In some instances, starting service may fail with "`localhost: rcmd: socket: Permission denied`" error; in that case we need to inform `pdsh` to use `ssh` instead of `rsh` by setting system variable `PDSH_RCMD_TYPE` (Listing 3.29).

```
export PDSH_RCMD_TYPE=ssh

sbin/start-dfs.sh

sbin/stop-dfs.sh
```

Listing 3.29. Commands to start and stop HDFS cluster. *Source: Author's own*

After the HDFS cluster has started, we need to create a user directory inside (Listing 3.30). We need to create each folder separately, as this utility cannot automatically create parent folders if they do not exist.

```
bin/hdfs dfs -mkdir /user

bin/hdfs dfs -mkdir /user/adrian
```

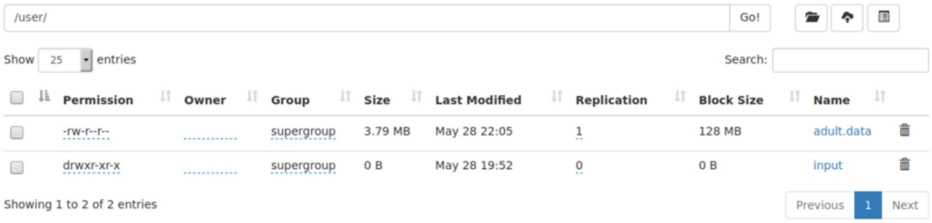
Listing 3.30. Creation of user folder. *Source: Author's own*

After the user folder is created we can create folders or upload files to that folder by omitting `/user/<username>` part of the path. For instance, we can create a new folder named indataput or put adult.data dataset into our user directory, by issuing the following command where first argument after `–put` is local path and the second one os path inside the cluster (Listing 3.31).

```
bin/hdfs dfs -mkdir data

bin/hdfs dfs -put adult.data .
```

Listing 3.31. Creation of data folder inside user directory and upload of local file to user directory on cluster. *Source: Author's own*

By default Apache Hadoop runs web pages with information about name nodes on port 9870. Name node web page provides also a graphical interface to see what files are stored on the cluster and provides a way to upload files (Fig. 3.10).



/user/ Go!

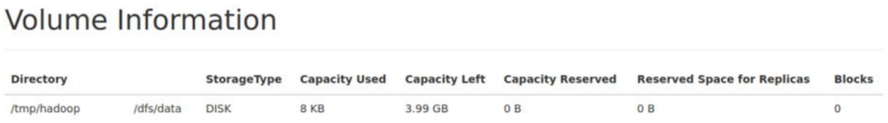
Show 25 entries Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	-rw-r--r--		supergroup	3.79 MB	May 28 22:05	1	128 MB	adult.data	🗑
□	drwxr-xr-x		supergroup	0 B	May 28 19:52	0	0 B	input	🗑

Showing 1 to 2 of 2 entries

Figure 3.10. Name node webpage showing files stored in user directory. *Source: Author’s own*

Data node webpage is available under port 9864 and provides access to various logs and information about the amount of data stored on the node (Fig. 3.11).



Volume Information

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/tmp/hadoop /dfs/data	DISK	8 KB	3.99 GB	0 B	0 B	0

Figure 3.11. Data node webpage showing filesystem usage. *Source: Author’s own*

To use files stored in HDFS cluster in Apache Spark environments, we have to only provide a path to a specific share in `hdfs://<name node ip>/<path to file>` format. If we wanted to use data stored on the HDFS cluster in the previous example, we could replace the code from the listing 3.21 with the code from the listing 3.32.

```
df = spark.read.load(
    'hdfs://localhost/user/user/adult.data',
    format='csv', sep=', ',
    header=False,    schema=data_schema
)
```

Listing 3.32. Usage of file stored on HDFS cluster inside Apache Spark function. *Source: Author’s own*

In the fourth example, we will try to cluster data from the public seeds dataset [3.6]. Data clustering is used when the data doesn’t contain any categorization in a form of labels by grouping data into sets depending on their proximity to the set center. While SparkSession creation is similar to previous examples, the data columns are separated by a variable tab number. Firstly, we read each row to one textual column for later processing (Listing 3.33).

```
df = spark.read.load('seeds_dataset.txt', format='text')
```

Listing 3.33. Import of file as plain text. *Source: Author’s own*

Now we can split each textual row on each tab character or multiple of them. We can store the prepared split function in a variable for more concise code later on (Listing 3.34).

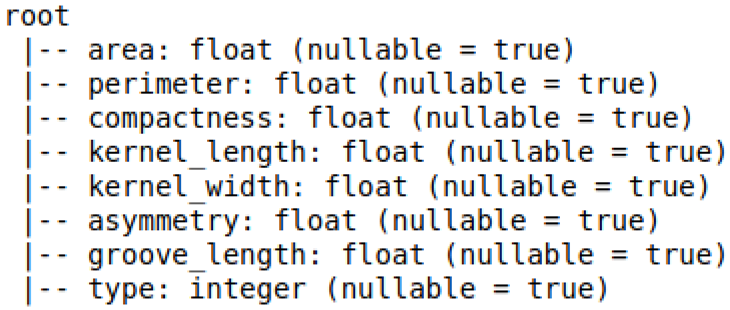
```
from pyspark.sql.functions import split

split_function = split('value', '\t+')
```

Listing 3.34. Prepared split function to split string on each cluster of tabs. *Source: Author's own*

Now we can create a new DataFrame, where we select each item from the split list, change name for each column with `alias` method and cast their types from string to float, or integer for seed type (Listing 3.35) (Fig. 3.12).

```
data = df.select(

    sp_f.getItem(0).alias('area').cast(FloatType()),

    sp_f.getItem(1).alias('perimeter').cast(FloatType()),

    sp_f.getItem(2).alias('compactness').cast(FloatType()),

    sp_f.getItem(3).alias('kernel_length').cast(FloatType()),

    sp_f.getItem(4).alias('kernel_width').cast(FloatType()),

    sp_f.getItem(5).alias('asymmetry').cast(FloatType()),

    sp_f.getItem(6).alias('groove_length').cast(FloatType()),

    sp_f.getItem(7).alias('type').cast(IntegerType()

)
```

Listing 3.35. Creation of new DataFrame from split column. *Source: Author's own*



```
root
 |-- area: float (nullable = true)
 |-- perimeter: float (nullable = true)
 |-- compactness: float (nullable = true)
 |-- kernel_length: float (nullable = true)
 |-- kernel_width: float (nullable = true)
 |-- asymmetry: float (nullable = true)
 |-- groove_length: float (nullable = true)
 |-- type: integer (nullable = true)
```

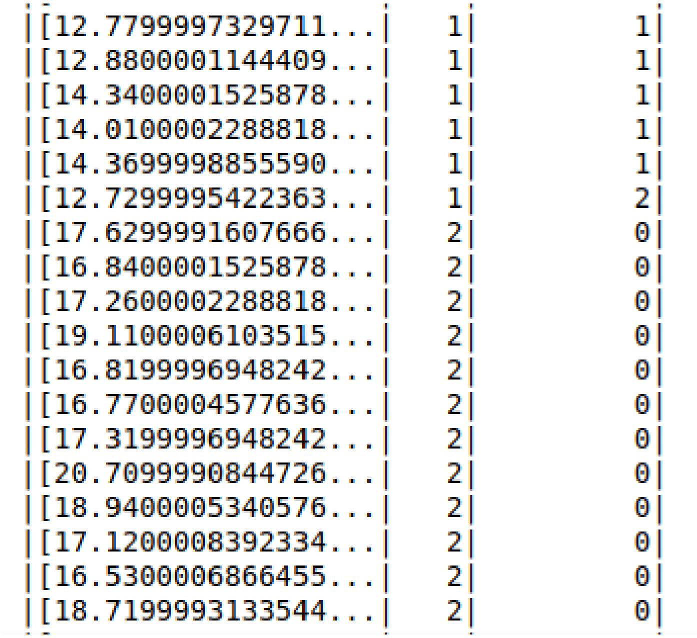
Figure 3.12. Schema of split data. *Source: Author's own*

When we have data ready we can use `VectorAssember` to combine all columns except type into one. After the data was prepared, we create a KMeans object with k set to the number of total types of wheat seeds in this dataset, which equals 3 (Listing 3.36).

```
from pyspark.ml.clustering import KMeans

kmeans = KMeans(featuresCol='features', k=3)

model = kmeans.fit(data)

results = model.transform(data)
```

Listing 3.36. Creation and training of K Means model. *Source: Author's own*

This particular the dataset contains labels, so we can compare visually how well the model assigned each row to one of the three clusters (Fig. 3.13). Please bear in mind that the labels numbers and estimated cluster numbers may differ, however the correlation between the two columns is clearly visible.



```
|[12.7799997329711...|   1|          1|
|[12.8800001144409...|   1|          1|
|[14.3400001525878...|   1|          1|
|[14.0100002288818...|   1|          1|
|[14.3699998855590...|   1|          1|
|[12.7299995422363...|   1|          2|
|[17.6299991607666...|   2|          0|
|[16.8400001525878...|   2|          0|
|[17.2600002288818...|   2|          0|
|[19.1100006103515...|   2|          0|
|[16.8199996948242...|   2|          0|
|[16.7700004577636...|   2|          0|
|[17.3199996948242...|   2|          0|
|[20.7099990844726...|   2|          0|
|[18.9400005340576...|   2|          0|
|[17.1200008392334...|   2|          0|
|[16.5300006866455...|   2|          0|
|[18.7199993133544...|   2|          0|
```

Figure 3.13. Fragment of results DataFrame, with features, type and prediction columns. *Source: Author's own*

However, in most cases the data processed by K Means algorithm, we do not have any labels, so we cannot compare how well our model behaves. We can access computed centers of each cluster and Apache Spark provides us with ClusteringEvaluator which computes the silhouette measure with help of the squared Euclidean distance (Listing 3.37).

```
print(model.clusterCenters())

from pyspark.ml.evaluation import ClusteringEvaluator

evaluator = ClusteringEvaluator()

silhouette = evaluator.evaluate(results)
print(silhouette)
```

Listing 3.37. Evaluation of trained K Means model. *Source: Author's own*

In our case the silhouette was equal to 0.66.

3.4. SUMMARY

The third chapter was devoted to the Apache Hadoop ecosystem with emphasis on Apache Spark.

From the description of core Hadoop projects and Spark framework to usage examples for both Spark and HDFS technologies. Included examples showed how to set up a single-node Hadoop Distributed File System, and how to manipulate data pySpark and create machine learning models.

3.5. EXERCISES

1. What is the advantage of Apache spark over Apache MapReduce?
2. Why DataFrame is recommended over RDD?
3. Can Apache HDFS run on a singular computer?
4. How can we determine the quality of a linear regression model?
5. Search Apache Spark documentation for available regression algorithms.
6. Can clustering be applied to unlabeled data?
7. Where is the data stored in HDFS?
8. Search Apache Hadoop documentation for information about providing name node resilience.
9. What are the ways that data stored in HDFS can be accessed?
10. What is the use for OneHotEncoder class?

3.6. REFERENCES

3.6.1. BIBLIOGRAPHY

[3.1] Estrada R., Ruiz I. (2016), *Big Data, Big Challenges*, In: *Big Data SMACK*, Apress, Berkeley, CA.

[3.2] Mishra R.K. (2018), *The Era of Big Data, Hadoop, and Other Big Data Processing Frameworks*, In: *PySpark Recipes*, Apress, Berkeley, CA.

[3.3] Quinto B. (2018), *Next-Generation Big Data*, In: *Next-Generation Big Data*, Apress, Berkeley, CA.

3.6.2. INTERNET REFERENCES

[3.4] http://archive.ics.uci.edu/ml/datasets/Iris

[3.5] http://archive.ics.uci.edu/ml/datasets/Adult

[3.6] https://archive.ics.uci.edu/ml/datasets/seeds

[3.7] https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

Closing remarks

Data is not information, information is not knowledge, knowledge is not understanding, understanding is not wisdom.

C. Stoll

The goal of this book was to provide an introduction to big data. The contents of the book have been divided into the following main parts: introduction, four chapters, closing remarks, references and appendix.

Chapter one was devoted to the theory of big data, while the next two chapters are related with practical aspects of big data – Chapter 2 and Chapter 3. In closing remarks, we have shown the areas of interest which were not covered in the book, but they are worth to be considered exploring after finishing reading this book.

The most important chapter of this handbook was related to practical aspects of big data, with a brief description of Apache Hadoop ecosystem as well as a description of practical examples of the use of big data solutions.

We have assumed that examples are not just another way of explanation, but we treat them as the key way to understand the issues presented in this book. We encourage our readers to explore the literature references as well as move further, keeping in mind, that the topic of big data is developing dynamically. In simple words, it is worth not sticking with state-of-the-art provided in the chapters, but completing them with new progress in research work.

If we look around, we can notice that computer solutions are present in our everyday life. Previously, without the connection to the Internet and nowadays, in more and more formulating the network of connected devices, changing our homes into smart homes, our cities into smart cities and our villages into smart villages. Ubiquitous computing is inseparable from data. Data sources can be integrated, formulating data lakes, and data lakes are large storage repositories (big data).

It leads to a whole set of applications of big data technologies. For example, improving time use measurement with big data, enabling big data at manufacturing processes application of big data in banking (e.g., credit-granting procedure). There exist applications of big data in libraries or in an analysis of campus life, use of big data in ocean's observation, use in healthcare or in driver state monitoring

Moreover, we are living in the world in which we explore the potential of cloud computing and machine learning. We may start our day with the question, which will be directed to our intelligent assistant, for example Amazon Alexa, Microsoft Cortana and Google Assistant. We deliver the voice question in a given language, that question needs to be processed (speech-to-text) and further processed with the use of natural language processing algorithms and finally, the answer shall be formulated and provided to us. It is so simple from a user perspective and so complex and fascinating from a technical point of view.

It would be great to consider exploring the area of machine learning and cloud computing.

This is the first edition of this book. It would be more than welcome if you decide to share your insights, ideas or spottederrors, thereby helping me to prepare a better quality of this book in the next edition. This book is equipped with additional electronic materials, which are published on Github platform. In the appendix located in the closing part of the book, you can find more details about it. Please follow presented below URL to find more: https://github.com/adriank-ps/bigdatabook-v1.

This book has been written for teaching purposes related to a course of study entitled: "Cognitive Technologies". This project is funded by Polish National Agency for Academic Exchange. Hereby, once again I would like to express my appreciation to prof. Aleksandra Kuzior for offering me the possibility to become a member of the project team.

REFERENCES

1. Akhtar, Aleem, *Role of Apache Software Foundation in Big Data Projects*, arXiv:2005.02829 [cs], May 2020, arXiv.org, http://arxiv.org/abs/2005.02829.
2. Akin, Ozgun at al., *Enabling Big Data Analytics at Manufacturing Fields of Farplas Automotive*, arXiv:2004.11682 [cs], April 2020, arXiv.org, http://arxiv.org/abs/2004.11682.
3. Albert Y. Zomaya and Sherif Sakr (2017), *Handbook of Big Data Technologies*, Springer.
4. Almasaari, Shakir A., *Securing Big Data systems, A cybersecurity management discussion*, arXiv:1912.08191 [cs], December 2019, arXiv.org, http://arxiv.org/abs/1912.08191.
5. Aziz, Fayeem at al.,. *Big Data in IoT Systems*, arXiv:1905.00490 [cs], April 2019, arXiv.org, http://arxiv.org/abs/1905.00490.
6. Azmoodeh A., Dehghantanha A. (2020), *Big Data and Privacy: Challenges and Opportunities*, In: Choo K.K., Dehghantanha A. (eds.), *Handbook of Big Data Privacy*, Springer, Cha.
7. Barua S., Begum S., Ahmed M.U. (2016), *Driver's State Monitoring: A Case Study on Big Data Analytics*, In: Ahmed M., Begum S., Raad W. (eds.), *Internet of Things Technologies for HealthCare*, HealthyIoT 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 187, Springer, Cham.
8. Bautista E., Whitney C., Davis T. (2016), *Big Data Behind Big Data*, In: Arora R. (eds.), *Conquering Big Data with High Performance Computing*, Springer, Cham.
9. Bohlouli, Mahdi et al., *Towards an Integrated Platform for Big Data Analysis*, arXiv:2004.13021 [cs], April 2020, arXiv.org, http://arxiv.org/abs/2004.13021.
10. Chaki S. (2015), *Pillar No. 8: Big Data Components*, In: *Enterprise Information Management in Practice*, Apress, Berkeley, CA.
11. Che D., Safran M., Peng Z. (2013), *From Big Data to Big Data Mining: Challenges, Issues, and Opportunities*, In: Hong B., Meng X., Chen L., Winiwarter W., Song W. (eds.), *Database Systems for Advanced Applications*, DASFAA 2013. Lecture Notes in Computer Science, vol. 7827, Springer, Berlin, Heidelberg.
12. Chebbi I., Boulila W., Farah I.R. (2015), *Big Data: Concepts, Challenges and Applications*, In: Núñez M., Nguyen N., Camacho D., Trawiński B. (eds.), *Computational Collective Intelligence*, Lecture Notes in Computer Science, vol 9330, Springer, Cham.
13. Cheng X., Fang L., Yang L., Cui S. (2018), *Mobile Big Data*, In: *Mobile Big Data*, Wireless Networks, Springer, Cham.
14. Christen M., Blumer H., Hauser C., Huppenbauer M. (2019), *The Ethics of Big Data Applications in the Consumer Sector*, In: Braschler M., Stadelmann T., Stockinger K. (eds.), *Applied Data Science*, Springer, Cham.
15. Cremonesi M., Bellini C., Bian B., Canali L., Dimakopoulos V., Elmer P., Fisk I., Girone M., Gutsche O., Hoh S.Y. et al. (2019), *Using Big Data Technologies for HEP Analysis*, EPJ Web of Conferences, 214, 06030.
16. Curry E. (2016), *The Big Data Value Chain: Definitions, Concepts, and Theoretical Approaches*, In: Cavanillas J., Curry E., Wahlster W. (eds.), *New Horizons for a Data-Driven Economy*, Springer, Cham.
17. Dai H.N., Wang H., Xu G., Wan J., Imran M. (2019), *Big data analytics for manufacturing internet of things: opportunities, challenges and enabling technologies*, Enterprise Information Systems, 1-25.
18. D'Alconzo A., Drago I., Morichetta A., Mellia M., Casas P. (2019), *A Survey on Big Data for Network Traffic Monitoring and Analysis*, IEEE Transactions on Network and Service Management, 16(3), 800-813.

19. Demchenko Y., Ngo C., de Laat C., Membrey P., Gordijenko D. (2014), *Big Security for Big Data: Addressing Security Challenges for the Big Data Infrastructure*, In: Jonker W., Petković M. (eds.), *Secure Data Management*, SDM 2013. Lecture Notes in Computer Science, vol. 8425, Springer, Cham.

20. Divya Zion G., Tripathy B.K. (2020), *Comparative Analysis of Tools for Big Data Visualization and Challenges*, In: Anouncia S., Gohel H., Vairamuthu S. (eds.), *Data Visualization*, Springer, Singapore.

21. Edward S.G., Sabharwal N. (2015), *Big Data*, In: *Practical MongoDB*, Apress, Berkeley, CA.

22. Elgendy N., Elragal A. (2014), *Big Data Analytics: A Literature Review Paper*, In: Perner P. (eds.), *Advances in Data Mining*, Applications and Theoretical Aspects. ICDM 2014. Lecture Notes in Computer Science, vol 8557, Springer, Cham.

23. Elgendy N., Elragal A. (2014), *Big Data Analytics: A Literature Review Paper*, In: Perner P. (eds.), *Advances in Data Mining*, Applications and Theoretical Aspects. ICDM 2014. Lecture Notes in Computer Science, vol 8557, Springer, Cham.

24. Estrada R., Ruiz I. (2016), *Big Data, Big Challenges*, In: *Big Data SMACK*, Apress, Berkeley, CA.

25. Furht B., Villanustre F. (2016), *Introduction to Big Data*, In: *Big Data Technologies and Applications*, Springer, Cham.

26. Gaitanou P., Garoufallou E., Balatsoukas P. (2014), *The Effectiveness of Big Data in Health Care: A Systematic Review*, In: Closs S., Studer R., Garoufallou E., Sicilia MA. (eds.), *Metadata and Semantics Research*, MTSR 2014. Communications in Computer and Information Science, vol 478, Springer, Cham.

27. Gorodetsky V. (2014), *Big Data: Opportunities, Challenges and Solutions*, In: Ermolayev V., Mayr H., Nikitchenko M., Spivakovsky A., Zholtkevych G. (eds.), *Information and Communication Technologies in Education, Research, and Industrial Applications*, ICTERI 2014. Communications in Computer and Information Science, vol 469, Springer, Cham.

28. Gronwald K.D. (2017), *Business Intelligence (BI) and Big Data Analytics (Big Data)*, In: *Integrated Business Information Systems*, Springer, Berlin, Heidelberg.

29. Guo L., Xu W., Li H., Zhang S., Zhao D. (2016), *The Application of Big Data Technology in the Field of Combat Simulation Data Management*, In: Zhang L., Song X., Wu Y. (eds.), *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems*, AsiaSim 2016, SCS Autumn Sim 2016. Communications in Computer and Information Science, vol 645, Springer, Singapore.

30. Helbing D. (2015), *Big Data – A Powerful New Resource for the Twenty-first Century*, In: *Thinking Ahead – Essays on Big Data, Digital Revolution, and Participatory Market Society*, Springer, Cham.

31. Hou W., Guo P., Guo L. (2015), *Networking Big Data: Definition, Key Technologies and Challenging Issues of Transmission*, In: Wang Y., Xiong H., Argamon S., Li X., Li J. (eds.), *Big Data Computing and Communications*, BigCom 2015. Lecture Notes in Computer Science, vol 9196, Springer, Cham.

32. Hussein E., Sadiki R., Jafta Y., Sungay M.M., Ajayi O., Bagula A. (2020), *Big Data Processing Using Hadoop and Spark: The Case of Meteorology Data*, In: Zitouni R., Agueh M., Houngue P., Soude H. (eds.), *e-Infrastructure and e-Services for Developing Countries*, AFRICOMM 2019. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 311, Springer, Cham.

33. Jaraba Navas P.C., Guacaneme Parra Y.C., Rodríguez Molano J.I. (2016), *Big Data Tools: Hadoop, MongoDB and Weka*, In: Tan Y., Shi Y. (eds.), *Data Mining and Big Data*, DMBD 2016. Lecture Notes in Computer Science, vol 9714, Springer, Cham.

34. Jedlitschka A. (2017), *Analyzing the Potential of Big Data*, In: Felderer M., Méndez Fernández D., Turhan B., Kalinowski M., Sarro F., Winkler D. (eds.), *Product-Focused Software Process Improvement*, PROFES 2017. Lecture Notes in Computer Science, vol. 10611, Springer, Cham.

35. Khanan A., Abdullah S., Mohamed A.H.H.M., Mehmood A., Ariffin K.A.Z. (2019), *Big Data Security and Privacy Concerns: A Review*, In: Al-Masri A., Curran K. (eds.), *Smart Technologies and Innovation for a Sustainable Future*, Advances in Science, Technology & Innovation (IEREK Interdisciplinary Series for Sustainable Development), Springer, Cham.

36. Lake P., Crowther P. (2013), *Big Data*, In: *Concise Guide to Databases*, Undergraduate Topics in Computer Science, Springer, London.

37. Lake P., Drake R. (2014), *Introducing Big Data*, In: *Information Systems Management in the Big Data Era*, Advanced Information and Knowledge Processing, Springer, Cham.

38. Lee N. (2014), *Consumer Privacy in the Age of Big Data*, In: *Facebook Nation*, Springer, New York, NY.

39. Li Q. et al. (2019), *Big Data Architecture and Reference Models*, In: Debruyne C., Panetto H., Guédria W., Bollen P., Ciuciu I., Meersman R. (eds.), *On the Move to Meaningful Internet Systems: OTM 2018 Workshops*, OTM 2018. Lecture Notes in Computer Science, vol. 11231, Springer, Cham.

40. Liu Y., Qiu M., Liu C., Guo Z. (2016), *Big Data in Ocean Observation: Opportunities and Challenges*, In: Wang Y., Yu G., Zhang Y., Han Z., Wang G. (eds.), *Big Data Computing and Communications*, BigCom 2016. Lecture Notes in Computer Science, vol. 9784, Springer, Cham.

41. Luengo J., García-Gil D., Ramírez-Gallego S., García S., Herrera F. (2020), *Big Data Software*, In: *Big Data Preprocessing*, Springer, Cham.

42. Luengo J., García-Gil D., Ramírez-Gallego S., García S., Herrera F. (2020), *Big Data: Technologies and Tools*, In: *Big Data Preprocessing*, Springer, Cham.

43. Lyu F., Ren L., Du Y. (2017), *An Optimization Method for User Interface Components Based on Big Data*, In: Zhang L., Ren L., Kordon F. (eds.), *Challenges and Opportunity with Big Data. Monterey Workshop 2016*, Lecture Notes in Computer Science, vol 10228, Springer, Cham.

44. Mazumder S. (2016), *Big Data Tools and Platforms*, In: Yu S., Guo S. (eds.), *Big Data Concepts, Theories, and Applications*, Springer, Cham.

45. Mazumder S. (2016), *Big Data Tools and Platforms*, In: Yu S., Guo S. (eds.), *Big Data Concepts, Theories, and Applications*, Springer, Cham.

46. Mikalef P., Pappas I.O., Krogstie J. et al. (2018), *Big data analytics capabilities: a systematic literature review and research agenda*, Inf Syst E-Bus Manage 16, 547-578.

47. Mishra R.K. (2018), *The Era of Big Data, Hadoop, and Other Big Data Processing Frameworks*, In: *PySpark Recipes*, Apress, Berkeley, CA.

48. Mohanty S., Jagadeesh M., Srivatsa H. (2013), A*pplication Architectures for Big Data and Analytics*, In: *Big Data Imperatives*, Apress, Berkeley, CA.

49. Nafchi, Mohsen Aghabozorgi, Maryam Aghabozorgi Nafchi, *Challenges and Opportunities of Big Data in Healthcare Mobile Applications*, arXiv:1906.10166 [cs], June 2019, arXiv.org, http://arxiv.org/abs/1906.10166.

50. Nataliya, Shakhovska et al., *Generalized formal model of big data*, arXiv:1905.03061 [cs], maj 2019, arXiv.org, http://arxiv.org/abs/1905.03061.

51. Obitko M., Jirkovský V., Bezdíček J. (2013), *Big Data Challenges in Industrial Automation*, In: Mařík V., Lastra J.L.M., Skobelev P. (eds.), *Industrial Applications of Holonic and*

Multi-Agent Systems, Lecture Notes in Computer Science, vol 8062, Springer, Berlin, Heidelberg.

52. Odegua, Rising, Festus Ikpotokin, *DataSist: A Python-based library for easy data analysis, visualization and modeling*, arXiv:1911.03655 [cs], January 2020, arXiv.org, http://arxiv.org/abs/1911.03655.
53. Olendorf R., Wang Y. (2017), *Big Data in Libraries*, In: Suh S., Anthony T. (eds.), *Big Data and Visual Analytics*, Springer, Cham.
54. Óskarsdóttir M., Bravo C., Sarraute C., Vanthienen J., Baesens B. (2019), *The value of big data for credit scoring: Enhancing financial inclusion using mobile phone data and social network analytics*, Applied Soft Computing, 74, 26-39.
55. Pastor-Escuredo, David, *Ethics in the digital era*, arXiv:2003.06530 [cs], March 2020, arXiv.org, http://arxiv.org/abs/2003.06530.
56. Pormeister K. (2017), *The GDPR and Big Data: Leading the Way for Big Genetic Data?*, In: Schweighofer E., Leitold H., Mitrakas A., Rannenberg K. (eds.), *Privacy Technologies and Policy*, APF 2017. Lecture Notes in Computer Science, vol 10518, Springer, Cham.
57. Prabhu C., Chivukula A., Mogadala A., Ghosh R., Livingston L. (2019), *Big Data Analytics in Bio-informatics*, In: *Big Data Analytics: Systems, Algorithms, Applications*, Springer, Singapore.
58. Prabhu C., Chivukula A., Mogadala A., Ghosh R., Livingston L. (2019), *Big Data Analytics for Insurance*, In: *Big Data Analytics: Systems, Algorithms, Applications*, Springer, Singapore.
59. Quinto B. (2018), *Next-Generation Big Data*, In: *Next-Generation Big Data*, Apress, Berkeley, CA.
60. Radhika D., Aruna Kumari D. (2018), *Adding Big Value to Big Businesses: A Present State of the Art of Big Data, Frameworks and Algorithms*, In: Saini A., Nayak A., Vyas R. (eds.), *ICT Based Innovations*, Advances in Intelligent Systems and Computing, vol 653, Springer, Singapore.
61. Rafferty W., Rafferty L., Hung P.C.K. (2016), *Introduction to Big Data*, In: Hung P. (eds.), *Big Data Applications and Use Cases*, International Series on Computer Entertainment and Media Technology, Springer, Cham.
62. Rambousek A., Parkin H., Horak A. (2018), *Software Tools for Big Data Resources in Family Names Dictionaries Names*, 66(4), 246-255.
63. Ramya Devi R., Vijaya Chamundeeswari V. (2020), *Triple DES: Privacy Preserving in Big Data Healthcare*, Int J. Parallel Prog 48, 515-533, https://doi.org/10.1007/s10766-018-0592-8.
64. Rehman, Arshia et al., *Leveraging Big Data Analytics in Healthcare Enhancement: Trends, Challenges and Opportunities*, arXiv:2004.09010 [cs, stat], April 2020, arXiv.org, http://arxiv.org/abs/2004.09010.
65. Rodríguez-Molano J.I., Contreras-Bravo L.E., López-Santana E.R. (2018), *Big Data Tools for Smart Cities*, In: Tan Y., Shi Y., Tang Q. (eds.), *Data Mining and Big Data*, DMBD 2018. Lecture Notes in Computer Science, vol 10943, Springer, Cham.
66. Rosenfeld, Ariel et al., *Big Data Analytics and AI in Mental Healthcare*, arXiv:1903.12071 [cs], marzec 2019, arXiv.org, http://arxiv.org/abs/1903.12071.
67. Sangeetha S., Sudha Sadasivam G. (2019), *Privacy of Big Data: A Review*, In: Dehghantanha A., Choo KK. (eds.), *Handbook of Big Data and IoT Security*, Springer, Cham.
68. Santos A.F.C., Teles Í.P., Siqueira O.M.P., de Oliveira A.A. (2018), *Big Data: A Systematic Review*, In: Latifi S. (eds.), *Information Technology – New Generations*, Advances in Intelligent Systems and Computing, vol 558, Springer, Cham.

69. Sawant N., Shah H. (2013), *Big Data Deployment Patterns*, In: *Big Data Application Architecture Q & A*, Apress, Berkeley, CA.

70. Shi Y., Quan P. (2020), *Big Data Analysis: Theory and Applications*, In: Lirkov I., Margenov S. (eds.) *Large-Scale Scientific Computing*, LSSC 2019. Lecture Notes in Computer Science, vol 11958, Springer, Cham.

71. Spraker K. (2018), *Difficulties Implementing Big Data: A Big Data Implementation Study*, In: Kurosu M. (eds.), *Human-Computer Interaction. Interaction in Context*, HCI 2018. Lecture Notes in Computer Science, vol 10902, Springer, Cham.

72. Steinmann M. et al. (2015), *Embedding Privacy and Ethical Values in Big Data Technology*, In: Matei S., Russell M., Bertino E. (eds.), *Transparency in Social Media*, Computational Social Sciences, Springer, Cham.

73. Trevisan, Martino, *Big Data for Traffic Monitoring and Management*, arXiv:1902.11095 [cs], February 2019. arXiv.org, http://arxiv.org/abs/1902.11095.

74. Wadkar S., Siddalingaiah M. (2014), *Motivation for Big Data*, In: *Pro Apache Hadoop*, Apress, Berkeley, CA.

75. Wani M.A., Jabin S. (2018), *Big Data: Issues, Challenges, and Techniques in Business Intelligence*, In: Aggarwal V., Bhatnagar V., Mishra D. (eds.), *Big Data Analytics*, Advances in Intelligent Systems and Computing, vol 654, Springer, Singapore.

76. Whang KY. (2018), *Recent Trends of Big Data Platforms and Applications*, In: Trujillo J. et al. (eds.), *Conceptual Modeling*, ER 2018. Lecture Notes in Computer Science, vol 11157, Springer, Cham.

77. Yang, Zongkai et al., *Evolution Features and Behavior Characters of Friendship Networks on Campus Life*, arXiv:2004.06266 [physics, stat], April 2020, arXiv.org, http://arxiv.org/abs/2004.06266.

78. Yoo S., Choi K., Lee M. (2014), *Business Ecosystem and Ecosystem of Big Data*, In: Chen Y. et al. (eds.), *Web-Age Information Management*, WAIM 2014. Lecture Notes in Computer Science, vol 8597, Springer, Cham.

79. Yoshida N. (2015), *A Large Sky Survey Project and the Related Big Data Analysis*, In: Chu W., Kikuchi S., Bhalla S. (eds.), *Databases in Networked Information Systems*, DNIS 2015. Lecture Notes in Computer Science, vol 8999, Springer, Cham.

80. Yue Y., Li D. (2019), *Big Data Challenges of FAST*, Lecture Notes in Computer Science, 6-9.

81. Zeni, Mattia et al., *Improving time use measurement with personal big data collection – the experience of the European Big Data Hackathon 2019*, arXiv:2004.11940 [cs], April 2020. arXiv.org, http://arxiv.org/abs/2004.11940.

82. Zhu, Dingju, *Big Data based Research on Mechanisms of Sharing Economy Restructuring the World*, arXiv:2001.08926 [econ, q-fin], January 2020. arXiv.org, http://arxiv.org/abs/2001.08926.

Appendixes

A.1. Source code (full listings)

```
from pyspark.sql import SparkSession

spark = SparkSession.Builder().appName('iris').getOrCreate()

from pyspark.sql.types import StructType, StructField,
DecimalType, StringType

data_schema = StructType([

     StructField('sepal_l', DecimalType(scale=1), False),

 StructField('sepal_w', DecimalType(scale=1), False),

 StructField('petal_l', DecimalType(scale=1), False),

 StructField('petal_w', DecimalType(scale=1), False),

 StructField('class', StringType(), False)])

# [dataset](http://archive.ics.uci.edu/ml/datasets/Iris)

df = spark.read.csv('iris.data', header=False,
schema=data_schema)

df.show(3)

from pyspark.ml.feature import StringIndexer

indexer = StringIndexer(inputCol='class',
outputCol='class_index')

from pyspark.ml.feature import VectorAssembler

features_assembler = VectorAssembler(inputCols=['sepal_l',
'sepal_w', 'petal_l', 'petal_w'], outputCol='features')

from pyspark.ml.classification import LogisticRegression

regressor =  LogisticRegression(featuresCol='features',
labelCol='class_index')

from pyspark.ml import Pipeline

pipeline = Pipeline(stages=[indexer, features_assembler])

trained_pipeline = pipeline.fit(df)

final_data = trained_pipeline.transform(df)

final_data.show(3)

train_data, test_data = final_data.randomSplit([0.7, 0.3])

regression_model = regressor.fit(train_data)

results = regression_model.transform(test_data)

results.printSchema()

from pyspark.ml.evaluation import
MulticlassClassificationEvaluator
```

```
evaluator =
MulticlassClassificationEvaluator(predictionCol='prediction',
labelCol='class_index', metricName='accuracy')

summary = evaluator.evaluate(results)

print(summary)

results.select(['class_index','rawPrediction','probability','
prediction']).show(3, truncate=False)
```

Listing A.1.1. Code of first example from chapter 3. *Source: Author's own*

```
from pyspark.sql import SparkSession

spark = SparkSession.Builder().appName('adult').getOrCreate()

# [Dataset](http://archive.ics.uci.edu/ml/datasets/Adult)

from pyspark.sql.types import StructType, StructField,
DecimalType, StringType, IntegerType

data_schema = StructType([

 StructField('age', IntegerType()),

 StructField('workclass', StringType()),

 StructField('fnlwgt', IntegerType()),

 StructField('education', StringType()),

 StructField('education-num', IntegerType()),

 StructField('marital-status', StringType()),

 StructField('occupation', StringType()),

 StructField('relationship', StringType()),

 StructField('race', StringType()),

 StructField('sex', StringType()),

 StructField('capital-gain', IntegerType()),

 StructField('capital-loss', IntegerType()),

 StructField('hours-per-week', IntegerType()),

 StructField('native-country', StringType()),

 StructField('labels', StringType())

])

df = spark.read.load('adult.data', format='csv', sep=', ',
header=False, schema=data_schema)

df.printSchema()

data = df.replace('?', None)

data = data.na.drop(how='any')
```

```
data.printSchema()

from pyspark.ml.feature import StringIndexer, OneHotEncoder

workclass_indexer = StringIndexer(inputCol='workclass',
outputCol='workclass_index')

workclass_encoder = OneHotEncoder(inputCol='workclass_index',
outputCol='workclass_encoded')

education_indexer = StringIndexer(inputCol='education',
outputCol='education_index')

education_encoder = OneHotEncoder(inputCol='education_index',
outputCol='education_encoded')

marital_status_indexer = StringIndexer(inputCol='marital-
status', outputCol='marital-status_index')

marital_status_encoder = OneHotEncoder(inputCol='marital-
status_index', outputCol='marital-status_encoded')

occupation_indexer = StringIndexer(inputCol='occupation',
outputCol='occupation_index')

occupation_encoder =
OneHotEncoder(inputCol='occupation_index',
outputCol='occupation_encoded')

relationship_indexer = StringIndexer(inputCol='relationship',
outputCol='relationship_index')

relationship_encoder =
OneHotEncoder(inputCol='relationship_index',
outputCol='relationship_encoded')

race_indexer = StringIndexer(inputCol='race',
outputCol='race_index')

race_encoder = OneHotEncoder(inputCol='race_index',
outputCol='race_encoded')

sex_indexer = StringIndexer(inputCol='sex',
outputCol='sex_index') #only 2  possibilities

native_country_indexer = StringIndexer(inputCol='native-
country', outputCol='native-country_index')

native_country_encoder = OneHotEncoder(inputCol='native-
country_index', outputCol='native-country_encoded')

labels_indexer = StringIndexer(inputCol='labels',
outputCol='labels_index') # only 2 possibilities

from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(inputCols=['age',

  'workclass_encoded',

  'fnlwgt',
  'education_encoded',
  'education-num',
  'marital-status_encoded',
  'occupation_index',
  'relationship_encoded',
  'race_encoded',
  'sex_index',
  'capital-gain',
  'capital-loss',
  'hours-per-week',
  'native-country_encoded',
  'labels_index'], outputCol='features')
from pyspark.ml import Pipeline
pipeline = Pipeline(stages=[workclass_indexer,
 workclass_encoder,
 education_indexer,
 education_encoder,
 marital_status_indexer,
 marital_status_encoder,
 occupation_indexer,
 occupation_encoder,
 relationship_indexer,
 relationship_encoder,
 race_indexer,
 race_encoder,
 sex_indexer,
 native_country_indexer,
 native_country_encoder,
 labels_indexer,
 assembler
])
prepared_data_pipeline = pipeline.fit(data)
prepared_data = prepared_data_pipeline.transform(data)
prepared_data.printSchema()
```

```
prepared_data = prepared_data.select(['features',
'labels_index'])

prepared_data.printSchema()

train_data, test_data = prepared_data.randomSplit([0.7, 0.3])

from pyspark.ml.classification import RandomForestClassifier,
DecisionTreeClassifier

rf_classifier =
RandomForestClassifier(featuresCol='features',
labelCol='labels_index', numTrees=15)

model = rf_classifier.fit(train_data)

prediction = model.transform(test_data)

from pyspark.ml.evaluation import
MulticlassClassificationEvaluator

mc_evaluator = MulticlassClassificationEvaluator(

 predictionCol='prediction',

 labelCol='labels_index',

 metricName="accuracy")

results = mc_evaluator.evaluate(prediction)

from pyspark.ml.evaluation import
BinaryClassificationEvaluator

bc_evaluator =
BinaryClassificationEvaluator(rawPredictionCol='rawPrediction
', labelCol='labels_index', metricName='areaUnderROC')

bc_results = bc_evaluator.evaluate(prediction)

print(bc_results)
```

Listing A.1.2. Code of second example from chapter 3. *Source: Author’s own*

```
from pyspark.sql import SparkSession

spark = SparkSession.Builder().appName('adult').getOrCreate()

# [Dataset](http://archive.ics.uci.edu/ml/datasets/Adult)

from pyspark.sql.types import StructType, StructField,
DecimalType, StringType, IntegerType

data_schema = StructType([

 StructField('age', IntegerType()),

 StructField('workclass', StringType()),

 StructField('fnlwgt', IntegerType()),

 StructField('education', StringType()),

 StructField('education-num', IntegerType()),
 StructField('marital-status', StringType()),
 StructField('occupation', StringType()),
 StructField('relationship', StringType()),
 StructField('race', StringType()),
 StructField('sex', StringType()),
 StructField('capital-gain', IntegerType()),
 StructField('capital-loss', IntegerType()),
 StructField('hours-per-week', IntegerType()),
 StructField('native-country', StringType()),
 StructField('labels', StringType())
])
df =
spark.read.load('hdfs://localhost/user/user/input/adult.data'
, format='csv', sep=', ', header=False, schema=data_schema)
df.printSchema()
```

Listing A.1.3. Code of third example from chapter 3. *Source: Author's own*

```
from pyspark.sql import SparkSession
spark =
SparkSession.builder.appName('clustering').getOrCreate()
from pyspark.sql.types import StructType, StructField,
FloatType, IntegerType, DoubleType
df = spark.read.load('seeds_dataset.txt', format='text')
from pyspark.sql.functions import split
sp_f = split('value', '\t+')
data =
df.select(sp_f.getItem(0).alias('area').cast(FloatType()),
 sp_f.getItem(1).alias('perimeter').cast(FloatType()),
 sp_f.getItem(2).alias('compactness').cast(FloatType()),
 sp_f.getItem(3).alias('kernel_length').cast(FloatType()),
 sp_f.getItem(4).alias('kernel_width').cast(FloatType()),
 sp_f.getItem(5).alias('asymmetry').cast(FloatType()),
 sp_f.getItem(6).alias('groove_length').cast(FloatType()),
 sp_f.getItem(7).alias('type').cast(IntegerType())
 )
from pyspark.ml.feature import VectorAssembler
```

```
assembler = VectorAssembler(inputCols=['area',
  'perimeter',
  'compactness',
  'kernel_length',
  'kernel_width',
  'asymmetry',
  'groove_length'], outputCol='features')
data = assembler.transform(data)
data.printSchema()
data = data.select('features', 'type')
from pyspark.ml.clustering import KMeans
kmeans = KMeans(featuresCol='features', k=3)
model = kmeans.fit(data)
results = model.transform(data)
print(model.clusterCenters())
from pyspark.ml.evaluation import ClusteringEvaluator
evaluator = ClusteringEvaluator()
silhouette = evaluator.evaluate(results)
print(silhouette)
```

Listing A.1.4. Code of fourth example from chapter 3. *Source: Author’s own*

A.2. Big data landscape

Big data landscape (infrastructure) – part 1

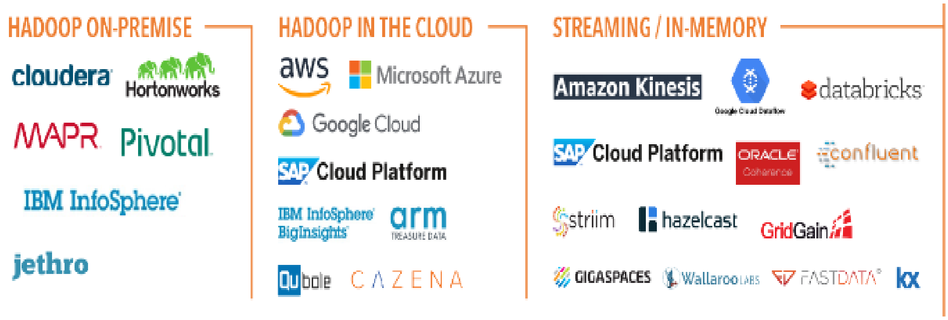


Figure A.2.1. Big data landscape (infrastructure) – part 1 – block 1.
Hadoop on-premise. Hadoop in the cloud. Streaming / in-memory.
Source: https://bit.ly/bigdata2020-landscape

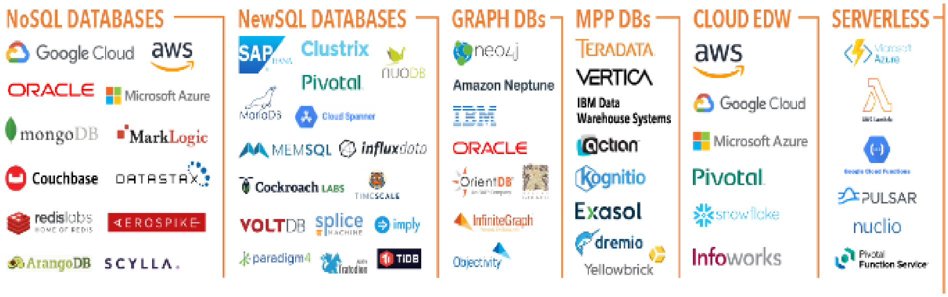


Figure A.2.2. Big data landscape (infrastructure) – part 1 – block 2.
NoSQL databases. NewSQL Databases. Graph Databases. MPP Databases. Cloud Enterprise Data Warehouses. Serverless. Source: https://bit.ly/bigdata2020-landscape

Big data landscape (infrastructure) – part 2

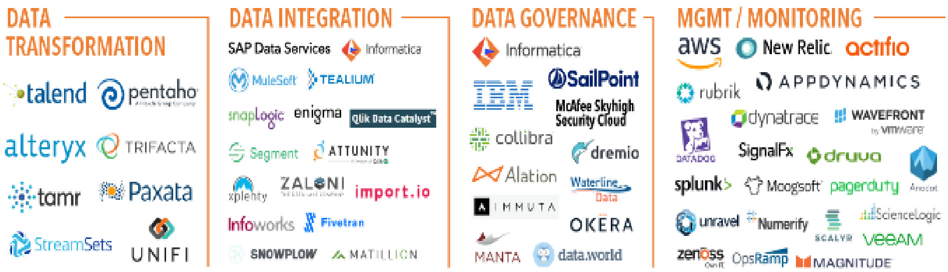


Figure A.2.3. Big data landscape (infrastructure) – part 2 – block 1.
Data transformation. Data integration. Data governance. Management (monitoring).
Source: https://bit.ly/bigdata2020-landscape

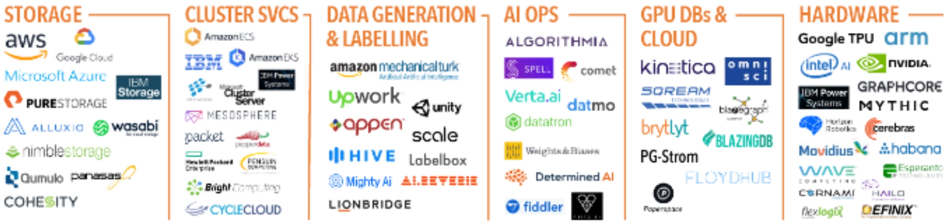


Figure A.2.4. Big data landscape (infrastructure) – part 2 – block 2.
Storage. Cluster services. Data generation and labelling. AI Operations. GPU databases and cloud. Hardware. Source: https://bit.ly/bigdata2020-landscape

Big data landscape (analytics and machine intelligence) – part 1

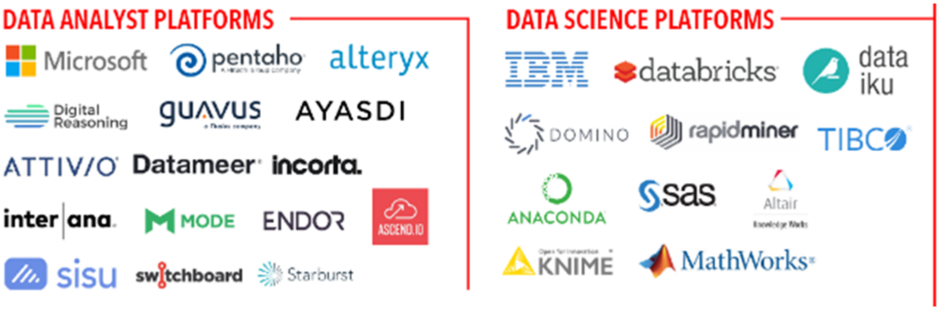


Figure A.2.5. Big data landscape (analytics and machine intelligence) – part 1 – block 1.
Data analyst platforms. Data science platforms. Source: https://bit.ly/bigdata2020-landscape

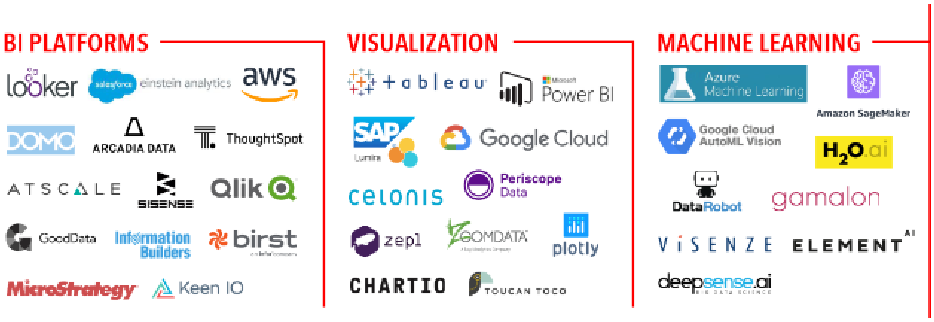


Figure A.2.6. Big data landscape (analytics and machine intelligence) – part 1 – block 2.
Business intelligence platforms. Visualization. Machine learning.
Source: https://bit.ly/bigdata2020-landscape

Big data landscape (analytics and machine intelligence) – part 2

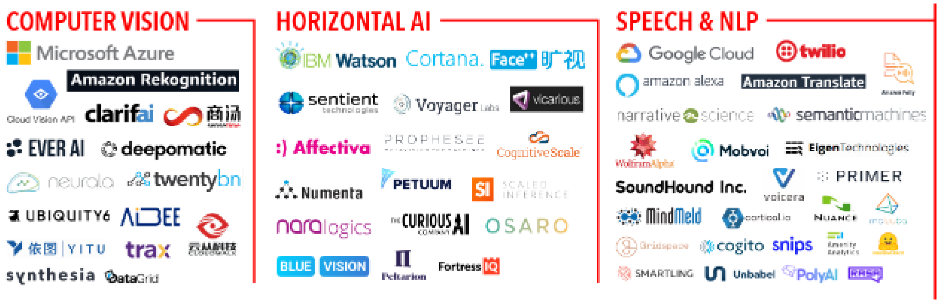


Figure A.2.7. Big data landscape (analytics and machine intelligence) – part 2 – block 1.
Computer vision. Horizontal AI. Speech and natural language processing.
Source: https://bit.ly/bigdata2020-landscape

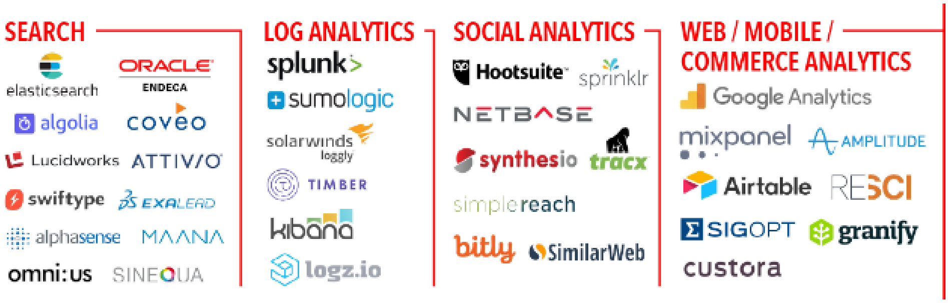


Figure A.2.8. Big data landscape (analytics and machine intelligence) – part 2 – block 2.
Search. Log analytics. Social analytics. Web, mobile, commerce analytics.
Source: https://bit.ly/bigdata2020-landscape

Big data landscape (applications) – part 1

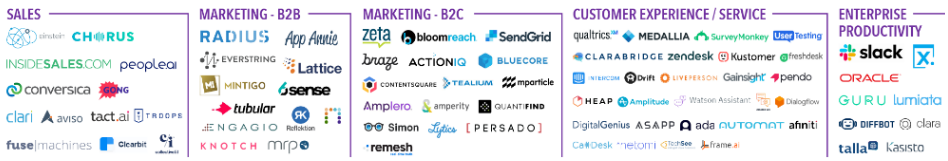


Figure A.2.9. Big data landscape (applications) – part 1 – block 1.
Sales. Marketing (B2B). Marketing (B2C). Customer experience (service). Enterprise productivity.
Source: https://bit.ly/bigdata2020-landscape

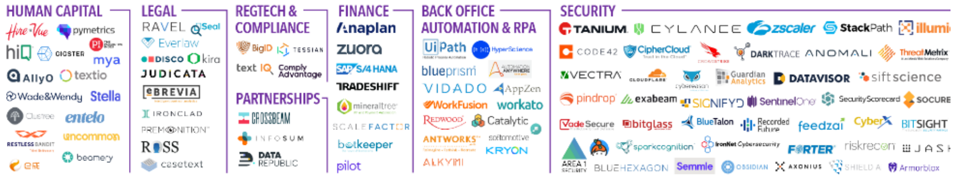


Figure A.2.10. Big data landscape (applications) – part 1 – block 2.
Human capital. Legal. Regulations and compliance. Back office automation and robotic process automation. Security, Source: https://bit.ly/bigdata2020-landscape

Big data landscape (applications) – part 2

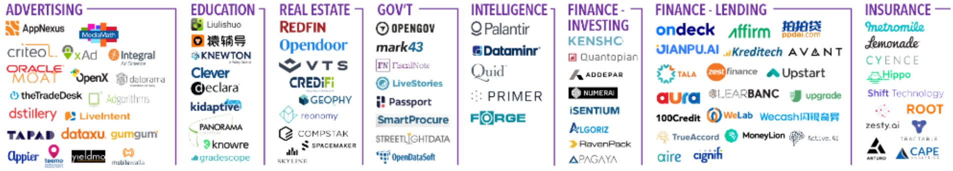


Figure A.2.11. Big data landscape (applications) – part 2 – block 1.
Advertising. Education. Real estate. Government. Intelligence. Finance-investing. Finance – lending. Insurance. Enterprise productivity. Source: https://bit.ly/bigdata2020-landscape

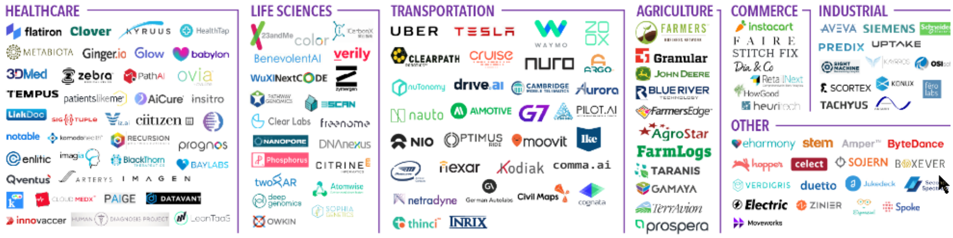


Figure A.2.12. Big data landscape (applications) – part 2 – block 2.
Healthcare. Life sciences. Transportation. Agriculture. Commerce. Industrial.
Source: https://bit.ly/bigdata2020-landscape

Big data landscape (open source) – part 1

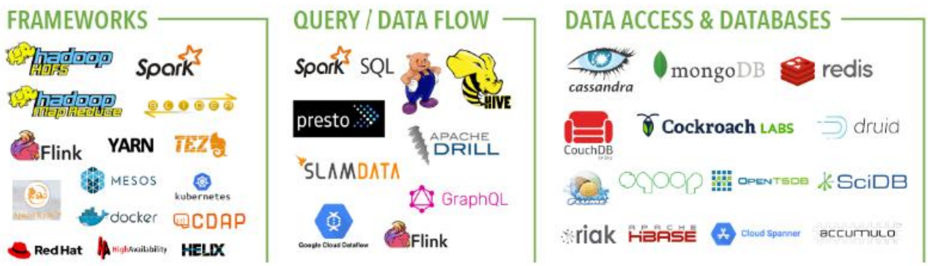


Figure A.2.13. Big data landscape (open source) – part 1 – block 1.
Big data landscape (open source) – part 1 – block #1: Frameworks, query (data flow). Data access and databases. Source: https://bit.ly/bigdata2020-landscape

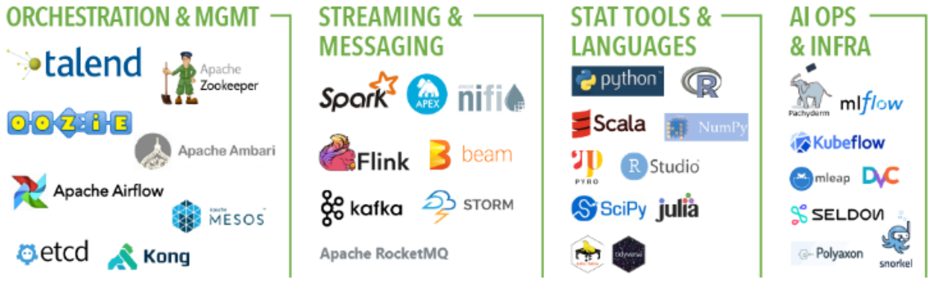


Figure A.2.14. Big data landscape (open source) – part 1 – block 2.
Orchestration and management. Streaming and messaging. Statistics tools and languages. AI Operations and infrastructure. Source: https://bit.ly/bigdata2020-landscape

Big data landscape (open source) – part 2

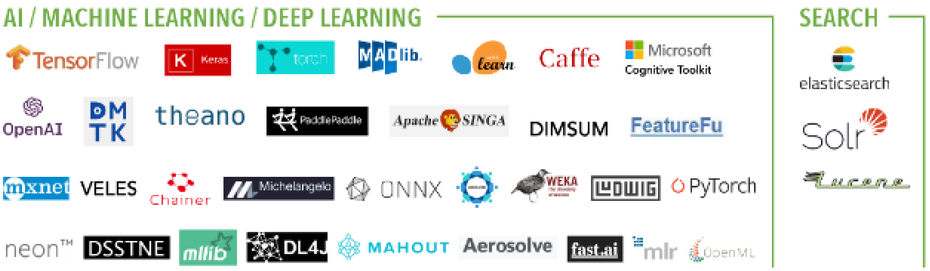


Figure A.2.15. Big data landscape (open source) – part 2 – block 1.
Artificial intelligence. Machine learning. Deep learning. Search.
Source: https://bit.ly/bigdata2020-landscape

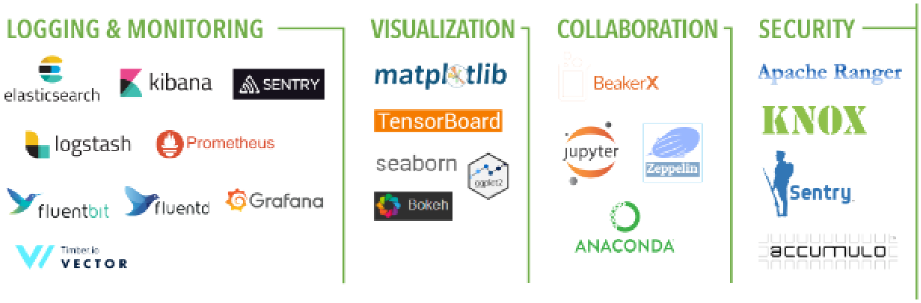


Figure A.2.16. Big data landscape (open source) – part 2 – block 2.
Logging and monitoring. Visualization. Collaboration. Security.
Source: https://bit.ly/bigdata2020-landscape

Big data landscape (data sources, API, data resources)

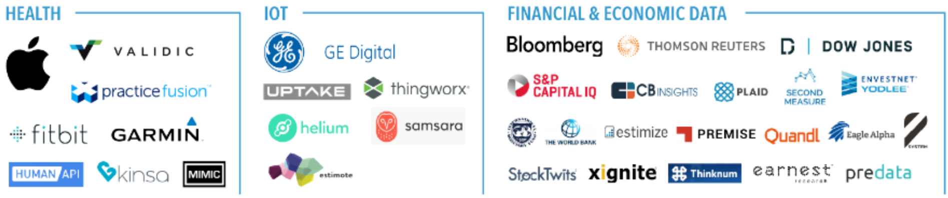


Figure A.2.17. Big data landscape (data sources, API, data resources) – block 1.
Health. IoT. Financial and economic data. Source: https://bit.ly/bigdata2020-landscape

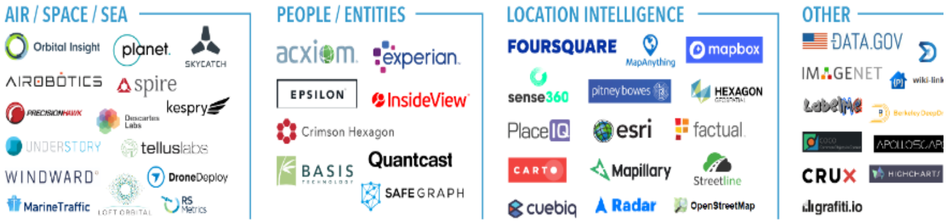


Figure A.2.18. Big data landscape (data sources, API, data resources) – block 2.
Air, space, sea. People, entities. Location intelligence. Other.
Source: https://bit.ly/bigdata2020-landscape

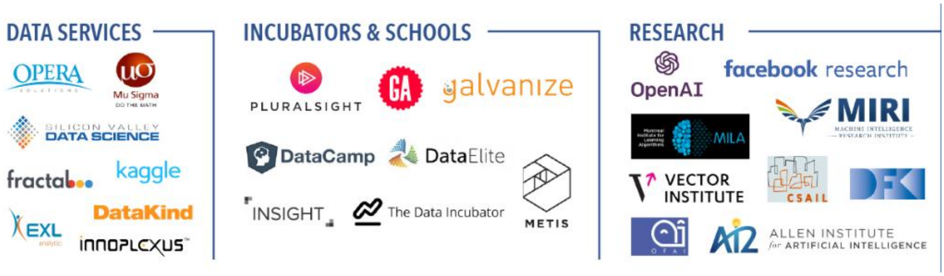


Figure A.2.19. Big data landscape (data sources, API, data resources) – block 3.
Data services. Incubators and schools. Research. Source: https://bit.ly/bigdata2020-landscape

Big data landscape (companies) – part 1

Table A.2.1. Big data landscape companies (infrastructure)

Name	Sub-category	URL
BlueData Software	Hadoop On-Premise	http://www.bluedata.com/
Cloudera	Hadoop On-Premise	http://cloudera.com/
Hortonworks	Hadoop On-Premise	http://hortonworks.com/
Altiscale	Hadoop in the Cloud	https://www.altiscale.com/
Amazon Elastic Map Reduce	Hadoop in the Cloud	https://aws.amazon.com/elasticmapreduce/
Cazena	Hadoop in the Cloud	https://www.cazena.com/
Amazon Kinesis	Streaming / In-Memory	https://aws.amazon.com/kinesis/
Amiato	Streaming / In-Memory	http://amiato.com/
Confluent	Streaming / In-Memory	http://www.confluent.io/
Aerospike	NoSQL Databases	http://www.aerospike.com/
Amazon DynamoDB	NoSQL Databases	https://aws.amazon.com/dynamodb/
ArangoDB	NoSQL Databases	https://www.arangodb.com/
Citus Data	NewSQL Databases	https://www.citusdata.com/
Clustrix	NewSQL Databases	http://www.clustrix.com/
CockroachDB	NewSQL Databases	http://www.cockroachlabs.com/
Amazon Neptune	Graph Databases	https://aws.amazon.com/neptune/
Aster Data	Graph Databases	http://www.teradata.com/Teradata-Aster/
Aurelius TitanDB	Graph Databases	http://dfkoz.com/ai-data-landscape/
Actian	MPP Databases	http://www.actian.com/
Dremio	MPP Databases	http://dremio.com/
Exasol	MPP Databases	http://www.exasol.com/en/
Amazon Redshift	Cloud EDW	https://aws.amazon.com/redshift/
Azure Data Lake	Cloud EDW	https://azure.microsoft.com/en-us/solutions/data-lake/
Google BigQuery	Cloud EDW	https://cloud.google.com/bigquery/
AWS Lambda	Serverless	https://aws.amazon.com/lambda/
Azure Functions	Serverless	https://azure.microsoft.com/en-us/services/functions/
Google Cloud Functions	Serverless	https://cloud.google.com/functions/
Alteryx	Data Transformation	http://www.alteryx.com/
Kalido	Data Transformation	http://kalido.com/
Paxata	Data Transformation	http://www.paxata.com/
Alooma	Data Integration	http://www.alooma.com/
Attunity	Data Integration	https://www.attunity.com/
Bedrock Data	Data Integration	http://www.bedrockdata.com/
Alation	Data Governance	https://alation.com/
BackOffice Associates	Data Governance	http://www.boaweb.com/
Collibra	Data Governance	https://www.collibra.com/
Actifio	Mgmt/Monitoring	http://www.actifio.com/
Amazon CloudWatch	Mgmt/Monitoring	https://aws.amazon.com/cloudwatch/
Anodot	Mgmt/Monitoring	http://www.anodot.com/

Alluxio	Storage	https://www.alluxio.com/
Amazon S3	Storage	https://aws.amazon.com/s3/
Amplidata	Storage	http://amplidata.com/
Amazon ECS	Cluster Services	https://aws.amazon.com/ecs/
Amazon EKS	Cluster Services	https://aws.amazon.com/eks/
Azure CycleCloud	Cluster Services	https://azure.microsoft.com/en-us/features/azure-cyclecloud/
AI.Reverie	Data Generation & Labelling	https://aireverie.com/
Amazon Mechanical Turk	Data Generation & Labelling	https://www.mturk.com/mturk/welcome
DataGen	Data Generation & Labelling	https://www.datagen.tech/
Algorithmia	AI Ops	https://algorithmia.com/
Comet.ml	AI Ops	https://www.comet.ml/
Datatron	AI Ops	https://www.datatron.com/
Blazegraph	GPU DBs & Cloud	https://www.blazegraph.com/
BlazingDB	GPU DBs & Cloud	https://blazingdb.com/
Brytlyt	GPU DBs & Cloud	https://www.brytlyt.com/
ARM	Hardware	https://www.arm.com/
Cerebras	Hardware	http://cerebras.net/
Cornami	Hardware	http://cornami.com/

Source: https://bit.ly/bigdata2020-landscape-raw

Big data landscape (companies) – part 2

Table A.2.2. Big data landscape companies (Analytics)

Name	Sub-category	More information
Alteryx	Data Analyst Platforms	http://www.alteryx.com/
Arimo	Data Analyst Platforms	http://arimo.com/
Ascend.io	Data Analyst Platforms	http://www.ascend.io/
Alpine Data Labs	Data Science Platforms	http://alpinedatalabs.com/
Anaconda (fka Continuum Analytics)	Data Science Platforms	https://www.anaconda.com/
Civis Analytics	Data Science Platforms	https://civisanalytics.com/
Amazon QuickSight	BI Platforms	https://aws.amazon.com/quicksight/
Arcadia Data	BI Platforms	http://www.arcadiadata.com/
AtScale	BI Platforms	http://www.atscale.com/
Actuate	Visualization	http://www.actuate.com/
Captain Dash	Visualization	http://www.captaindash.com/en/
Celonis	Visualization	http://www.celonis.de/
AlchemyAPI	Machine Learning	http://www.alchemyapi.com/
Amazon Sagemaker	Machine Learning	https://aws.amazon.com/sagemaker/
Azure ML Studio	Machine Learning	https://studio.azureml.net/
20BN	Computer Vision	https://20bn.com/
Aibee	Computer Vision	http://www.aibee.com/
Amazon Rekognition	Computer Vision	https://aws.amazon.com/rekognition/
Affectiva	Horizontal AI	https://www.affectiva.com/
Blue Vision Labs	Horizontal AI	http://www.bluevisionlabs.com/
Cortana Analytics	Horizontal AI	http://www.microsoft.com/en-us/server-cloud/cortana-analytics-suite/overview.aspx
Amazon Alexa	Speech & NLP	https://developer.amazon.com/alexa
Amazon Polly	Speech & NLP	https://aws.amazon.com/polly/
Amazon Translate	Speech & NLP	https://aws.amazon.com/translate/
Algolia	Search	https://www.algolia.com/
AlphaSense	Search	http://www.alpha-sense.com/
Attivio	Search	https://www.attivio.com/
Kibana	Log Analytics	https://www.elastic.co/products/kibana
Loggly	Log Analytics	https://www.loggly.com/
LogicMonitor	Log Analytics	https://www.logicmonitor.com/
Bitly	Social Analytics	https://bitly.com/
Bluefin labs	Social Analytics	https://bluefinlabs.com/
DataSift	Social Analytics	http://datasift.com/
Airtable	Web/Mobile/Commerce Analytics	https://airtable.com/
Amplitude	Web/Mobile/Commerce Analytics	https://amplitude.com/
Clavis Insight	Web/Mobile/Commerce Analytics	https://www.clavisinsight.com/

Source: https://bit.ly/bigdata2020-landscape-raw

Big data landscape (companies) - part 3

Table A.2.3. Big data landscape companies (Applications)

Name	Sub-category	URL
Aviso	Sales	http://www.aviso.com/
Chorus.ai	Sales	http://www.chorus.ai/
Clearbit	Sales	http://www.clearbit.com/
6sense	Marketing - B2B	https://6sense.com/
Alyce	Marketing - B2B	https://www.alyce.com/
App Annie	Marketing - B2B	https://www.appannie.com/
ActionIQ	Marketing - B2C	http://www.actioniq.com/
Amperity	Marketing - B2C	http://www.amperity.com/
Amplero	Marketing - B2C	https://www.amplero.com/
Ada	Customer Experience / Service	https://ada.support/
Afiniti	Customer Experience / Service	http://www.afiniti.com/
Amazon Lex	Customer Experience / Service	https://aws.amazon.com/lex/
Butter	Enterprise Productivity	https://www.butter.ai/
Clara Labs	Enterprise Productivity	https://claralabs.com/
Diffbot	Enterprise Productivity	http://www.diffbot.com/
AllyO	Human Capital	https://www.allyo.com/
Beamery	Human Capital	http://www.beamery.com/
Clustree	Human Capital	https://www.clustree.com/
Brightleaf	Legal	http://www.brightleaf.com/
Counselytics	Legal	http://counselytics.com/
CS Disco	Legal	http://www.csdisco.com/
BigID	RegTech & Compliance	http://www.bigid.com/
ComplyAdvantage	RegTech & Compliance	http://www.complyadvantage.com/
Tessian	RegTech & Compliance	https://www.tessian.com/
Adaptive Insights	Finance	http://www.adaptiveinsights.com/
Anaplan	Finance	https://www.anaplan.com/
Botkeeper	Finance	https://www.botkeeper.com/
Accelirate	Back Office Automation & RPA	https://www.accelirate.com/
Alkymi	Back Office Automation & RPA	https://www.alkymi.io/
AntWorks	Back Office Automation & RPA	https://www.ant.works/
Anomali	Security	http://www.anomali.com/
Area 1 Security	Security	https://www.area1security.com/
Armorblox	Security	https://www.armorblox.com/
33Across	Advertising	http://www.33across.com/
Adbrain	Advertising	http://www.adbrain.com/
Aggregate Knowledge	Advertising	https://www.neustar.biz/marketing-solutions
Clever	Education	https://clever.com/

Declara	Education	https://declara.com/
Gradescope	Education	http://www.gradescope.com
Compstak	Real Estate	https://compstak.com/
Credifi	Real Estate	https://www.credifi.com/
GeoPhy	Real Estate	http://www.geophy.com/
Enigma	Government	http://enigma.io/
FireStop	Government	http://www.firestopapp.com/
FiscalNote	Government	https://www.fiscalnote.com/
Dataminr	Intelligence	https://www.dataminr.com/
Forge.AI	Intelligence	https://www.forge.ai/
Palantir	Intelligence	https://www.palantir.com/
Addepar	Finance - Investing	https://addepar.com/
Algoriz	Finance - Investing	https://algoriz.com/
Clarity Money	Finance - Investing	http://www.claritymoney.com/
100credit	Finance - Lending	http://www.100credit.com/
Active AI	Finance - Lending	http://active.ai/
Affirm	Finance - Lending	https://www.affirm.com/
Arturo	Insurance	https://www.arturo.ai/
Cape Analytics	Insurance	http://www.capeanalytics.com/
Cyence	Insurance	http://www.cyence.net/
3DMED	Healthcare	http://www.3dmedcare.com/
AiCure	Healthcare	https://www.aicure.com/
Arterys	Healthcare	http://www.arterys.com/
23andMe	Life Sciences	https://www.23andme.com/
3scan	Life Sciences	http://www.3scan.com/
Atomwise	Life Sciences	http://www.atomwise.com/
Almotive	Transportation	http://www.aimotive.com
Argo AI	Transportation	https://www.argo.ai/
Aurora	Transportation	https://aurora.tech/
AgroStar	Agriculture	http://www.agrostar.in/
Aquabyte	Agriculture	https://www.aquabyte.no/
Blue River Tech	Agriculture	http://www.bluerivert.com/
Dia & Co	Commerce	https://www.dia.com/
Faire	Commerce	http://www.faire.com/
Heuritech	Commerce	https://www.heuritech.com/
Alluvium	Industrial	http://www.alluvium.io
Augury	Industrial	http://www.augury.com/
AVEVA	Industrial	https://www.aveva.com/
Amper Music	Other	https://www.ampermusic.com/
Boxever	Other	http://www.boxever.com/
Bytedance	Other	https://www.bytedance.com/

Source: https://bit.ly/bigdata2020-landscape-raw

Big data landscape (companies) - part 4

Table A.2.4. Big data landscape companies (Open source)

Name	Sub-category	URL
CDAP	Frameworks	http://cdap.io/
Docker	Frameworks	https://www.docker.com/
Flink	Frameworks	https://flink.apache.org/
Drill	Query/Data Flow	https://drill.apache.org/
Flink	Query/Data Flow	https://flink.apache.org/
Google Cloud DataFlow	Query/Data Flow	https://cloud.google.com/dataflow/
Accumulo	Data Access & Databases	https://accumulo.apache.org/
Cassandra	Data Access & Databases	http://cassandra.apache.org/
CockroachDB	Data Access & Databases	https://www.cockroachlabs.com/
Ambari	Orchestration & Management	https://ambari.apache.org/
Apache Airflow	Orchestration & Management	https://airflow.apache.org/
Apache Mesos	Orchestration & Management	https://mesos.apache.org/
Apex	Streaming & Messaging	http://incubator.apache.org/projects/apex.html
Beam	Streaming & Messaging	https://beam.apache.org/
Flink	Streaming & Messaging	https://flink.apache.org/
data.table	Stat Tools & Languages	https://github.com/Rdatatable/data.table/wiki
Julia	Stat Tools & Languages	https://julialang.org/
NumPy	Stat Tools & Languages	http://www.numpy.org/
DVC	AI Ops & Infra	https://dvc.org/
Kubeflow	AI Ops & Infra	https://www.kubeflow.org/
MLeap	AI Ops & Infra	http://mleap-docs.combust.ml/
Aerosolve	AI/Machine Learning/ Deep Learning	http://nerds.airbnb.com/aerosolve/
Caffe	AI/Machine Learning/ Deep Learning	http://caffe.berkeleyvision.org/
Caret	AI/Machine Learning/ Deep Learning	https://topepo.github.io/caret/
ElasticSearch	Search	https://github.com/elastic/elasticsearch
Lucene	Search	https://lucene.apache.org/
Solr	Search	http://lucene.apache.org/solr/
Elasticsearch	Logging & Monitoring	https://github.com/elastic/elasticsearch
Fluent Bit	Logging & Monitoring	https://fluentbit.io/
Fluentd	Logging & Monitoring	https://www.fluentd.org/
Bokeh	Visualization	https://bokeh.pydata.org/en/latest/docs/refe

		rence/models/widgets.tables.html
ggplot	Visualization	http://ggplot.yhathq.com/
Matplotlib	Visualization	https://matplotlib.org/
Anaconda	Collaboration	https://www.continuum.io/downloads
BeakerX	Collaboration	http://beakerx.com/
Jupyter	Collaboration	http://jupyter.org/
Accumulo	Security	https://accumulo.apache.org/
Knox	Security	https://knox.apache.org/
Ranger	Security	http://ranger.apache.org/

Source: https://bit.ly/bigdata2020-landscape-raw

Big data landscape (companies) – part 5

Table A.2.5. Big data landscape companies (Data sources)

Company	Sub-Category	URL
Apple	Health	http://www.apple.com/
Fitbit	Health	https://www.fitbit.com/
Garmin	Health	http://www.garmin.com/
Estimote	IOT	http://estimote.com/
GE Digital	IOT	https://www.ge.com/digital/
Helium	IOT	https://www.helium.com/
Bloomberg	Financial & Economic Data	http://www.bloomberg.com/
CapIQ	Financial & Economic Data	https://www.capitaliq.com/
CBInsights	Financial & Economic Data	http://www.cbinsights.com/
Airobotics	Air/Space/Sea	http://www.airobotics.co.il/
Airware	Air/Space/Sea	https://www.airware.com/
Descartes Labs	Air/Space/Sea	http://www.descarteslabs.com/
Acxiom	People/Entities	http://www.acxiom.com/
Basis Technology	People/Entities	http://www.basistech.com/
BlueKai	People/Entities	http://www.bluekai.com/
Carto	Location Intelligence	https://carto.com/
Cuebiq	Location Intelligence	http://www.cuebiq.com/
Esri	Location Intelligence	http://www.esri.com/
Apollo Scape	Other	http://apolloscape.auto/
Berkeley DeepDrive	Other	https://bdd-data.berkeley.edu/
COCO	Other	http://cocodataset.org/
Accenture	Data Services	https://www.accenture.com/
Caserta Concepts	Data Services	http://casertaconcepts.com/
DataKind	Data Services	http://www.datakind.org/
Data Elite	Incubators & Schools	http://www.dataeliteventures.com/
Data Science Workshops	Incubators & Schools	https://www.datascienceworkshops.com/
DataCamp	Incubators & Schools	https://www.datacamp.com/
Allen Institute	Research	http://allenai.org/
DFKI	Research	https://www.dfki.de/web/intelligent-solutions-for-the-knowledge-society?set_language=en
Facebook Research	Research	https://research.fb.com/category/facebook-ai-research-fair/

Source: https://bit.ly/bigdata2020-landscape-raw

European Funds
Knowledge Education Development

Republic of Poland

NAWA
POLISH NATIONAL AGENCY FOR ACADEMIC EXCHANGE

European Union
European Social Fund

Research, preparation of materials and preparation of the textbook were carried out under the project – grant no. PPI/KAT/2019/1/00015/U/00001 "Cognitive technologies – second-cycle studies in English" and were carried under the KATAMARAN program Polish National Agency for Academic Exchange (NAWA). The program is co-financed by the European Social Fund under the Knowledge Education Development Operational Program, a non-competition project entitled "Supporting the institutional capacity of Polish universities through the creation and implementation of international study programs" implemented under Measure 3.3. Internationalization of Polish higher education, specified in the application for project funding no. POWR.03.03.00-00-PN 16/18.

Wydawnictwo
TYGIEL

www.wydawnictwo-tygiel.pl

ISBN 978-83-66489-50-9

9 788366 489509