FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

# ASSIGNMENT OF MASTER'S THESIS

| | |
|---|---|
| **Title:** | Utilizing AI/ML methods for measuring data quality |
| **Student:** | Bc. Michael Mikuš |
| **Supervisor:** | Ing. Tomáš Pajurek |
| **Study Programme:** | Informatics |
| **Study Branch:** | Knowledge Engineering |
| **Department:** | Department of Applied Mathematics |
| **Validity:** | Until the end of summer semester 2020/21 |

## Instructions

Traditional data quality measurement methods need significant manual effort and domain expert experience. Apart from being a time-consuming process, these traditional methods are often based on manual definition of rules or thresholds and are therefore subject to human error. The utilization of AI/ML methods offers an alternative data-driven approach that might overcome some of the drawbacks of traditional methods.

1. Describe key aspects of data quality and review the state-of-the-art methods for monitoring and measuring data quality with special emphasis on AI/ML based approaches.
2. Experimentally compare traditional (non-AI) methods for measuring data quality with methods using AI/ML techniques.
3. Propose directions for further improvements in the application of AI/ML in data quality.

## References

Will be provided by the supervisor.

Ing. Karel Klouda, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague September 23, 2019

Czech Technical University in Prague
Faculty of Information Technology
Department of Applied Mathematics

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

# Utilizing AI/ML methods for measuring data quality

***Bc. Michael Mikuš***

Supervisor: Ing. Tomáš Pajurek

30th of July, 2020

# Acknowledgements

I would like to express sincere gratitude to my supervisor Ing. Tomáš Pajurek for his ideas, valuable advice, and continued mentoring over the past several years. I would also like to thank Ing. Tomáš Borovička for personal development in professional and study life, Datamole, s. r. o. for providing computational infrastructure for the experiments and my colleagues for their scientific support. Last but not least, I would like to thank my family, my parents, my sister and my friends for their emotional support during my studies. Finally, I would like to thank the Czech technical university in Prague that allowed me to discover the magic of education.

# Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on 30th of July, 2020 ……

Czech Technical University in Prague

Faculty of Information Technology



## Citation of this thesis

MIKUŠ, Michael. ***Utilizing AI/ML methods for measuring data quality***. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2020.

# Abstrakt

Kvalitní data jsou zásadní pro důvěryhodná rozhodnutí na datech založená. Značná část současných přístupů k měření kvality dat je spojena s náročnou, odbornou a časově náročnou prací, která vyžaduje manuální přístup k dosažení odpovídajících výsledků. Tyto přístupy jsou navíc náchylné k chybám a nevyužívají plně potenciál umělé inteligence (AI). Možným řešením je prozkoumat inovativní nové metody založené na strojovém učení (ML), které využívají potenciál AI k překonání těchto problémů.

Významná část práce se zabývá teorií kvality dat, která poskytuje komplexní vhled do této oblasti. V existující literatuře byly objeveny čtyři moderní metody založené na ML a byla navržena jedna nová metoda založená na autoenkodéru (AE).

Byly provedeny experimenty s AE a dolováním asociačních pravidel za pomoci metod zpracování přirozeného jazyka. Navrhované metody založené na AE prokázaly schopnost detekce potenciálních problémů s kvalitou dat na datasetech z reálného světa. Dolování asociačních pravidel dokázalo extrahovat byznys pravidla pro stanovený problém, ale vyžadovalo značné úsilí s předzpracováním dat. Alternativní metody nezaložené na AI byly také podrobeny analýze, ale vyžadovaly odborné znalosti daného problému a domény.

**Klíčová slova** datová kvalita, umělá inteligence, strojové učení, nástroje datové kvality, autoenkóder, asociační pravidla

# Abstract

High-quality data is crucial for trusted data-based decisions. A considerable part of current data quality measuring approaches is associated with expensive, expert and time-consuming work that includes manual effort to achieve adequate results. Furthermore, these approaches are prone to error and do not take full advantage of the AI potential. A possible solution is to explore ML-based state-of-the-art methods that are using the potential of AI to overcome these issues.

A significant part of the thesis deals with data quality theory which provides a comprehensive insight into the field of data quality. Four ML-based state-of-the-art methods were discovered in the existing literature, and one novel method based on Autoencoders (AE) was proposed.

Experiments with AE and Association Rule Mining using NLP were conducted. Proposed methods based on AE proved to detect potential data quality defects in real-world datasets. Association Rule Mining approach was able to extract business rules for a given business question, but the required significant preprocessing effort. Alternative non-AI methods were also analyzed but required reliance on expert and domain knowledge.

**Keywords** data quality, artificial intelligence, machine learning, data quality tools, autoencoder, association rules

# Contents

# List of Source codes

# List of Tables

# List of Figures

# Introduction

Managing data quality is often associated with expensive and expert work that includes manual effort to achieve adequate results. These financial and human resources costs are usually a substantial factor in why managing data quality is challenging and is perceived only as an auxiliary service by the companies.

In the worst case, the company purchases an expensive proprietary vendor lock-prone data quality tool, dedicates and trains several of its employees to learn how to use it. If the potential of these data quality tools is not fully realised the resulting business value is low.

To avoid not only the above-mentioned shortcomings in managing data quality but also to be able to look for an innovative AI approach which would reduce expert or user intervention into the field of data quality measurement, the key areas related to data quality were analyzed in this thesis – the nature of data (see Section 1.1 about Understanding data), essentials of data quality (see Section 1.2 about Understanding data quality) and the current data quality tools (see Section 1.3 about Existing data quality tools review).

The motivation of this work is to explore and experiment with state-of-the-art approaches for measuring data quality with special emphasis on AI/ML-based methods that have the potential to reduce the amount of manual work required.

This thesis uses the term data quality measurement in the context of various data quality activities (e.g. de-duplication, outliers detection, rules mining or monitoring as ongoing measurement) that are understood as an auxiliary part of the measurement.

In recent years, there has been a significant shift in the development of AI/ML-based methods, where their utilization helps to innovate across the industry, especially by shielding from human error and implementation of automation. Current general-purposed data quality tools do not take full advantage of the possibilities offered by these AI approaches. The reason for not using them on a broader scale may be the complexity of the application and utilization of these methods in the field of data quality.

For these reasons, the work analyzes potential of state-of-the-art approaches for measuring data quality with a focus on AI/ML (see Section 2 about Data quality measurement methods). For the experimental part of the work (see Section 3 about Experiments and Results), two AI/ML-based approaches to measuring data quality were selected – Autoencoders (see Section 3.3.1 about Experiment 1 – Autoencoder) and Association Rule Mining using NLP (see Section 3.3.2 about Experiment 2 – Association Rule Mining). Each of the approaches was compared with the nearest discovered complementary non-AI approach.

Chapter 1

# Theoretical framework

This chapter provides the reader with insight into key concepts of data (see Section 1.1 about Understanding data), data quality with a focus on measuring data quality (see Section 1.2 about Understanding data quality) and existing data quality tools (see Section 1.3 about Existing data quality tools review).

## 1.1 Understanding data

One of the most fundamental terms of this work is data (singular datum). Therefore the first section will focus on elementary theory and concepts associated with the term data (see Section 1.1.1 about Data in theoretical context).

The second part examines the diversity of data (see Section 1.1.2 about Heterogeneity in data) by dividing the data in different ways and finding the essential data properties that can play an important role in AI methods for measuring data quality.

At the end of the work, readers are acquainted with the effects of errors in the flow of data, which have significant consequences for data quality (see Section 1.1.3 about Quality consequences of Data life flow).

### 1.1.1 Data in theoretical context

Before discussing the quality of information (see Section 1.2.1 about Data quality in theoretical context), this section introduces one of the theoretical

views on data and information itself. Understanding data in a theoretical context is not trivial.

If we begin to explore the definition of information, we find that it may be associated with several explanations, depending on the specific area of application [65]. However, Luciano Floridy proposed the General Definition of Information (GDI)[1] as a part of his Philosophy of Information (see [23]). GDI provides a formal definition of information (from the semantic viewpoint) and a way in which information may be understood.

The first requisite GDI.1 of the definition points out that semantic information[2] consists of data, that can make things more complicated for a higher number of data [23]. In the second requisite GDI.2, the data are 'well-formed' – meaning that the data are assembled according to specific rules [23]. In the requisite GDI.3, the 'well-formed' data are 'meaningful' – meaning that the data are capable of being interpreted, translated or expressed differently and the last requisite GDI.4 requires an analysis of the nature of the true information (knowledge) [23].

Floridi's GDI definition provides us with a formal analysis of the information concept, with potentially direct value for information science [10]. Thought it is necessary to provide a class model (that it will be explained subsequently in the right away) to represent the relationships between data, information, knowledge, and (perhaps) wisdom to interconnect the above terms and place them in a broader context.

The class model is known as DIK(W) pyramid (consist of components – Data, Information, Knowledge, Wisdom) (see Figure 1.1). The main sources of the theory of the pyramid model DIKW can be found in the publications of Mortimer J. Adler[2], Milan Zeleny[63] and Ackoff[1]. DIKW model appears in several modifications [56] [10] and representations [54]. The modifications of the model omit or add components of the pyramid. Historical versions of the model do not include the data component, current versions are omitting and reducing the role of the wisdom component, and some include additional components (semantic metadata [56]). There are countless definitions of the

[1] in term of data + meaning

[2] understood as semantic content (simply information)

components, so only the general meaning of these components of the pyramid model will be given [65].

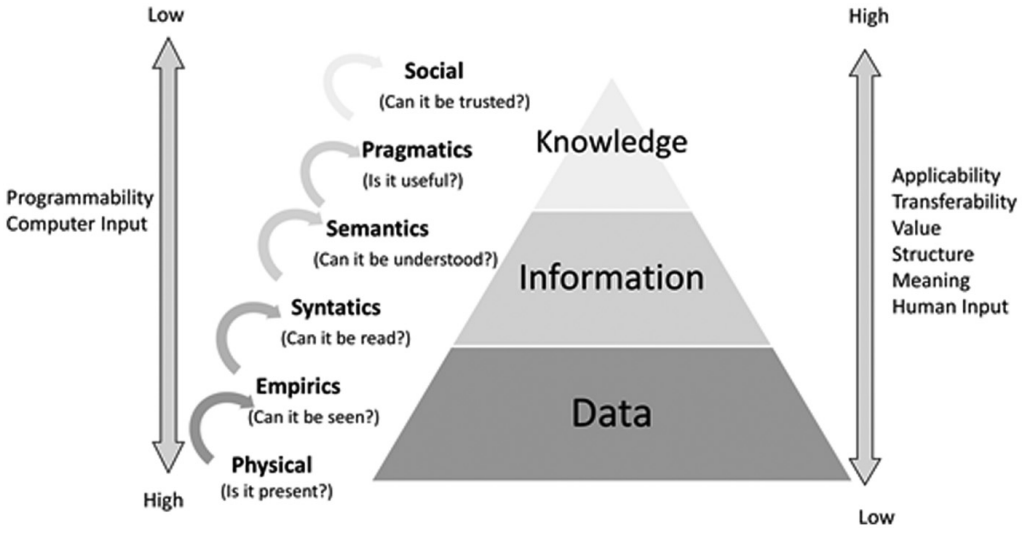


**Figure 1.1:** Semiotics levels and DIK hierarchy, adapted from Rowley and Burton-Jones [31].

In the context of the class model, DIKW pyramids generally start by holding data in any form (usable or not) that has no significance beyond its existence [11]. Information is data that is processed, organized or structured to answer elementary questions (e.g., ”who”, ”what”, ”where”, ”how many”, ”when”) and to understand a relationship of some sort (linked elements) [11]. At this point, the data can become useful for making decisions/actions. The knowledge (answers ”how” questions) and wisdom (appreciation of ”why”) component of DIKW is generally agreed to be an elusive concept which is difficult to define [54]. In general, knowledge seeks to understand patterns of a set of information (organized information) and wisdom seeks to understand fundamental principles embodied within the knowledge (applied knowledge) [11].

Therefore, it is necessary to ensure significant maturity of the initial components of the model in order to reduce the probability of making decisions based on bad data, which can be achieved by finding appropriate automated programmable procedures in the field of computer science. Programmability decreases

with a movement towards the top of the pyramid model [31]. This thesis is focused on those components that can be processed by computers. The question is what programmable procedures to use to improve the initial components of the model.

In the Figure 1.1, we can see semiotic[3] levels that correlate to DIK (i.e. empirics with physical signs, syntactics with data, semantics with information, and pragmatics with knowledge) and specific data quality dimensions (see more in Section 1.2.1) [31]. Semiotic theory concerns the use of symbols to convey knowledge [57]. Six levels are defined for symbol analysis:

1. ***Physical*** and ***empirical*** levels concern the physical media and use of the physical media for communication of symbols,
2. ***syntactic*** level concerns the structure of symbols and focuses on form rather than content,
3. ***semantic*** level concerns the meaning of symbols,
4. ***pragmatic*** level concerns the usage of symbols and is dependent on the task of the person using the data,
5. ***social*** level concerns the purpose of information in relation to social norms and social change [57] [31].

A detailed description of semiotic levels is beyond the scope of this work. Each semiotic level then addresses specific data quality skills and data quality issues associated with them [31] (see Section 1.2.1 about Data quality in theoretical context).

From the perspective of computer science, data represent a real word object, in a format that can be stored, retrieved, and elaborated by a software procedure, and communicated through a network. Data is very versatile in representing real objects in the world. In addition to information generated by the processing of computer data, there are other types of information that cannot be processed by a computer or can only be approximated (e.g. fragrance, taste). We will not deal with all of these types of information. The main focus of this thesis is on computer data. Computer data are processed by a computer's CPU and is stored

[3]Semiotics is the study of signs and symbols, their interpretation and use [31].

digitally in files and folders on the computer's hard disk in a defined format and are not spontaneously mutable[4] [18].

### 1.1.2 Heterogeneity in data

Heterogeneity is one of the major features of data and leads to several data issues. Data consists of various types (structured, semi-structured, unstructured), has different types of attributes (Qualitative and Quantitative), can be categorized differently (Dimensionality, Sparsity and Resolution), even by dataset characteristics (Record data, Graph data, Ordered data), and is stored in different diversions (Relational databases, Key-Value Stores or Column-Family Stores). Data heterogeneity can have a significant impact on data quality, as there are more options for how and where data can lose quality.

#### 1.1.2.1 Data areas

This section discusses the most common data areas to give the reader an insight into the current data technologies. Each of the data areas tries to investigate data problems that the data domain solves. There are countless of these areas, therefore only the most common areas will be discussed. Data areas may partially overlap. Some of the following areas will be mentioned without further description, as their extensive description is beyond the scope of this work.

***Web data*** is huge, widely-distributed, diverse, heterogeneous, semi-structured (see Section 1.1.2.2 about Types of data), linked (such as Linked data), redundant and dynamic information repository [4]. Web data consist of web content (e.g. text, images, records), web structure (e.g. hyperlinks or tags) and web usage (e.g. HTTP logs or app server logs) [4]. The discipline of obtaining information from web data is called Web mining (includes Web Content Mining, Web Structure Mining and Web Usage Mining) [4].

***Open data*** is data that is freely available to anyone in terms of its use, re-use, redistribution and rights to republish without restrictions from mechanisms of control (copyright, patents or other) [24]. Open data should be primary data (see Section 1.1.2.7 about Data categories), published in a timely manner and everyone must be able to use, re-use and redistribute them – there should be

[4]Computer data does not deteriorate over time or lose quality after being used multiple times [18].

no discrimination against fields of endeavour or against persons or groups [24]. For the data to be truly open, the following aspects should follow:

- data must be complete,
- data must be primary,
- data must be timely,
- data must be accessible,
- data must be machine processable and made online in persistent archives,
- access must be non-discriminatory,
- data formats must be non-proprietary,
- data license must be unrestricted and bear no usage costs,
- also data should be as accurate as possible [16].

Tim Berners-Lee designed a five-star rating scheme for open data according to the following criteria [29][28]:

- 1 Star: data is available on the Web (whatever format), but with an open license.
- 2 Stars: data is available as machine-readable structured data (e.g., Microsoft Excel instead of a scanned image of a table).
- 3 Stars: data is available as (2) but in a non-proprietary format (e.g., CSV instead of Excel).
- 4 Stars: data is available according to all the above, plus the use of open standards from the W3C (RDF and SPARQL) to identify things, so that people can link to it.
- 5 Stars: data satisfies all the above-mentioned points, including outgoing links to other people's data to provide context (LOD[5]).

***Linked data*** is a method for publishing structured interlinked data on the Web, building up on URIs, HTTP and RDF technologies [29]. The paragraphs dealing

[5]Linked Open Data

with linked data are drawn from the following source [29], unless otherwise stated.

These data are structured data in a machine-readable format that can be semantically queried. A standard mechanism for specifying the existence and meaning of connections between items described in this data is provided by the Resource Description Framework (RDF) that give us a way to describe real-world objects (people, locations, or abstract concepts) and their relationships with each other. Data enriched in this way is significantly more discoverable, and therefore more usable. The difference between a Classical Web and Semantic Web is that the Semantic Web connects objects and not just documents by saying what type of relationship it is (e.g. is-friend-of).

Linked data can link items between different data sources, therefore connect these sources into a single global data space via Web standards (URIs, HTTP, HTML) and a common data model (the Web of Data). The Web of Data (also referred to as Semantic Web) spans numerous topical domains, such as people, companies, and books, as well as an increasing volume of scientific and government data. Thus, the main goal is to share structured data (see Section 1.1.2.2 about Types of data) on a global scale. Linked data is not only intended for the public Web domain but can be applied to any data that can be linked by the described principle (e.g. private or personal).

Tim Berners-Lee put the linked data in the context of the following principles [29]:

1. Use URIs as names for things.
2. Use HTTP URIs, so that people can look up those names (interpreted, ”dereferenced”).
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL (query language is widely used for querying RDF data)).
4. Include links to other URIs, so that they can discover more objects.

With this approach, searching in data is more effective because of the interconnectedness of datasets coming from different and heterogeneous data sources.

One way to avoid heterogeneity is by advocating the reuse of terms from widely deployed vocabularies for describing common objects (e.g. people, companies, books). Therefore, when working with data for which an ontological dictionary already exists, it should be used (such as for Invoice, Airline or IoT)[6]. Another way is to make data self-descriptive as much as possible. Self-descriptive means that an application based on Linked data that discovers some data on the Web that is represented by a previously unknown dictionary should be able to find all the meta-information needed to translate the data into a representation that it understands and can process. Technically, every vocabulary term (such as RDFS, OWL, see following paragraph) links to its own definition and mapping between terms from different vocabularies in the form of RDF links should be published. These techniques lead to the discovery of meta-information for data integration.

"RDF provides a generic, abstract data model for describing resources using subject, predicate, object triples. However, it does not provide any domain-specific terms for describing classes of things in the world and how they relate to each other. This function is served by taxonomies, vocabularies and ontologies expressed in SKOS (Simple Knowledge Organization System), RDFS (the RDF Vocabulary Description Language, also known as RDF Schema) and OWL (the Web Ontology Language)." [29] A full discussion of RDF, SKOS, RDFS and OWL are beyond the scope of this thesis.

***Government data*** (e.g. economic statistics, land ownership, voting records of elected representatives and similar) is produced or commissioned by the government. Open government data relate to creating transparency, accountability and promoting democratic values [16].

***Big data*** is commonly described as more massive (no longer fits into the memory of a single machine) and complex datasets (structure, semi-structured, unstructured data) (see following Section 1.1.2.2) that require more sophisticated processing technologies (such as parallel and distributed computing framework Google's MapReduce, Hadoop and Apache Spark), streaming technologies (such as Apache Spark Streaming, Apache Storm, Apache Flink or Apache Samza) and storage technology (e.g. column-oriented databases stores)(see Section 1.1.2.6 about Diversity of data stores) than traditional approaches that were common

[6] http://ontologydesignpatterns.org/

before Big data era [16] [7]. Big Data is often explained and characterised using the 4V – volume, velocity, variety and veracity [16].

***Machine data*** includes data from areas as varied as IoT data or industrial systems sensor data, application programming interfaces (APIs), message queues, change events, cloud applications and other similar activities that are recorded. These are activities of machines, devices, customers, users, applications, servers, networks and similar in real-time. These activities create plenty of machine data in an array of unpredictable formats that are often ignored [12]. It is also often real-time or stream data. Machine data can help organizations troubleshoot problems, identify threats, and use machine learning to predict future issues [12]. Machine data are closely related to operational data (IT systems data such as application logs, metrics, event data and microservices applications)[12].

The following areas will be mentioned without further description – ***biological data*** (such as genomics data that analyse DNA [12], see Section 1.1.2.5) about General characteristics of datasets, ***multimedia data*** (see Figure 1.2) or ***social network data*** (study of human relationships by means of graph theory [4] – see Section 1.1.2.5 about General characteristics of datasets).

#### 1.1.2.2 Types of data

All data has structure of some sort and we have to count with a wide range of possible representations. Data involves storing structured, semi-structured, and unstructured multimedia data (see Figure 1.2). Most often we can encounter three types of data [9]:

***Structured data***, when each data element is bound to a rigid data structure (Schema, Data types) [9] [6]. Among the most well-known representatives of structured data is the relational table (such as relational databases). Thus, relational databases (without NoSQL/post-relational extensions) mostly process structured and formatted data [6].

***Semi-structured data*** (such as Twitter feeds, Facebook and YouTube postings), when data has a structure which has some degree of flexibility [9] with emphasis on human-readable aspect. Thus, data does not obey the fixed structure (schemaless) but contains tags or other markers used to identify certain elements within the data. The best-known formats include XML, JSON and

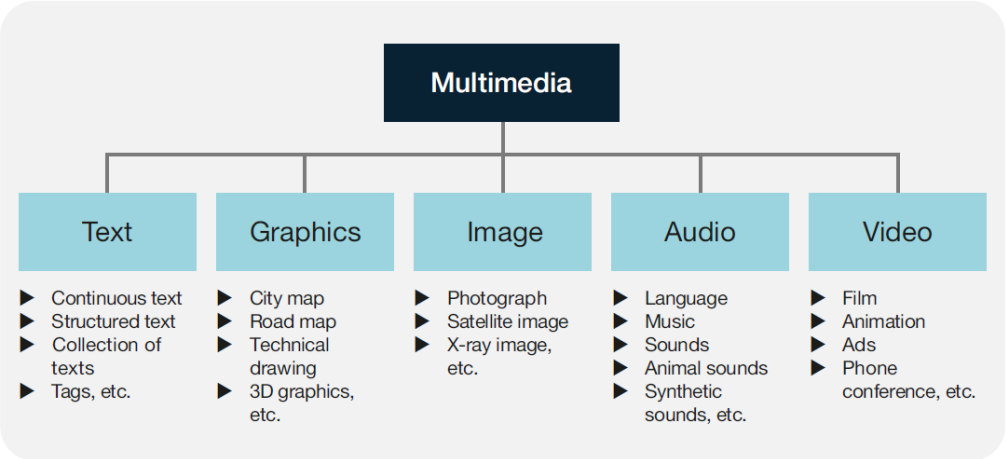


**Figure 1.2:** Variety of data sources [6].

HTML (in general web-related formats). This type of self-describing data is important in situations where the data collection is potentially valuable, but we cannot store it in a structured way, or we do not yet know its exact use.

***Unstructured data***, when data has no specific structure [9] and is very difficult or almost impossible for a computer to understand because of irregularities and ambiguities, such as text-heavy documents (such as books and journals – in general natural language) that can contain important unstructured facts. Unstructured data includes forms of images, videos and audio as well. Unstructured data became increasingly important only when storing it in larger amounts became financially viable for companies. Subsequently, new technologies such as the NoSQL databases have developed around unstructured and semi-structured data.

Most NoSQL systems (see Section 1.1.2.6 about Diversity of data stores) do not have an explicit database schema, since changes can happen at any time in the semistructured or unstructured data [6].

#### 1.1.2.3 Data attribute types

Generally, rows (instances, objects) in a dataset are characterized by the values of features, or attributes, that measures different aspects of the row [47]. Attributes can be divided into two basic qualitative and quantitative groups (see Figure 1.3). A useful and simple way to specify the type of an attribute is to identify the properties of numbers that correspond to underlying properties

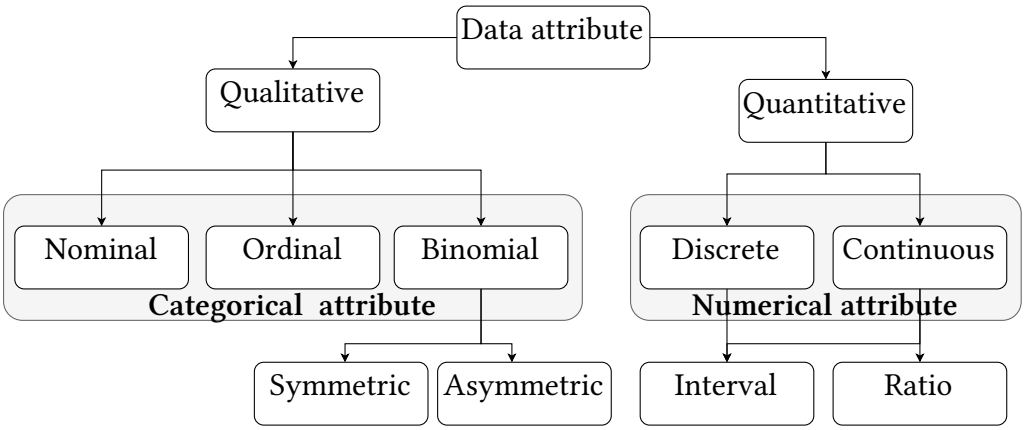


**Figure 1.3:** Data attributes types.

of the attribute [59]. The following properties (operations) of numbers are typically used to describe attributes:

1. ***Distinctness*** $=$ and $\neq$,
2. ***Order*** $<$, $\leq$, $>$, and $\geq$,
3. ***Addition*** $+$ and $-$,
4. ***Multiplication*** $*$ and $/$ [59].

Qualitative attribute refers to a value that cannot be expressed as a number, but we can group information according to their values. This group includes nominal, ordinal and binomial attributes that have a finite set of possibilities. Nominal attribute (e.g. race, eye color) values cannot be sorted or measured, and are used as labels or names. Ordinal attributes (e.g. performance) are the ones that can rank order of categories but cannot measure the distance between them [9]. A binomial attribute (e.g. a basic gender division) has allowed only two labels (disjunct sets) [9]. The binomial attributes are divided according to whether their values are equally important. For example, gender attribute values have the same weight symmetrically, but test results (e.g. pass/fail) are of different importance. The above mentioned attributes are classified into a group of categorical attributes.

Quantitative attribute refers to a value that can be expressed as a number or can be quantified [9]. This group includes discrete and continuous attributes. Discrete attribute (e.g. school grades, defects per hour) is countably infinite and continuous attribute (e.g. length measurements) is uncountably infinite [47][9]. Numerical attribute (discrete or continuous) can be interval-scaled or ration-scaled [47]. Interval attributes (e.g. temperatures in Celsius or Fahrenheit, calendar dates) have values that are not only ordered but also measured in fixed and equal units, and no natural starting point (e.g. zero) is specified – it is completely arbitrary [9]. Ratio attributes (e.g. length, temperature in Kelvin) compared to interval attributes are ones for which the measurement method inherently defines a zero point [9].

The Table 1.1 gives the descriptions of the main attribute types along with information about the statistical operations that are valid for each type and Table 1.2 explains their permissible (meaning-preserving) transformation (e.g. the meaning of a length attribute is unchanged if it is measured in meters instead of feet) [59].

"The statistical operations that make sense for a particular type of attribute are those that will yield the same results when the attribute is transformed using a transformation that preserves the attribute's meaning. To illustrate, the average length of a set of objects is different when measured in meters rather than in feet, but both averages represent the same length." [59]

The properties and operations of the attribute types are cumulative [59] (see detail description below). It means that any property or operation that is valid for nominal, ordinal, and interval attributes is also valid for ratio attributes[7] [59]. Specifically:

1. Nominal adheres to ***Distinctness***;
2. Ordinal adheres to ***Distinctness*** and ***Order***;
3. Interval adheres to ***Distinctness***, ***Order*** and ***Addition***;
4. Ratio adheres to ***Distinctness***, ***Order***, ***Addition*** and ***Multiplication*** [59].

[7]However, this does not mean that the operations appropriate for one attribute type are appropriate for the attribute types above it [59].

**Table 1.1:** Different attribute types [59].

| Attribute Type | | Description | Examples | Operations |
|---|---|---|---|---|
| **Categorical** (Qualitative) | ***Nominal*** | The values of a nominal attribute are just different names; i.e., nominal values provide only enough information to distinguish one object from another. $(=, \neq)$ | zip codes, employee ID numbers, eye color, gender | mode, entropy, contingency correlation, $\chi^2$ test |
| | ***Ordinal*** | The values of an ordinal attribute provide enough information to order objects. $(<, >)$ | hardness of minerals, $\{good, better, best\}$ , grades, street numbers | median, percentiles, rank correlation, run tests, sign tests |
| **Numeric** (Quantitative) | ***Interval*** | For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. $(+, -)$ | calendar dates, temperature in Celsius or Fahrenheit | mean, standard deviation Pearson's correlation, $t$ and $F$ tests |
| | ***Ratio*** | For ratio variables, both differences and ratios are meaningful. $(*, /)$ | temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current | geometric mean, harmonic mean, percent variation |

#### 1.1.2.4 General data characteristics

Before providing types of datasets, three general data characteristics will describe the data – Dimensionality, Sparsity and Resolution.

**Table 1.2:** Transformations that define attribute levels [59].

| Attribute Type | | Transformation | Comment |
|---|---|---|---|
| **Categorical** (Qualitative) | ***Nominal*** | Any one-to-one mapping, e.g., a permutation of values | If all employee ID numbers are reassigned, it will not make any difference. |
| | ***Ordinal*** | An order-preserving change of values, i.e., $new\_value = f(old\_value)$, where $f$ is a monotonic function. | An attribute encompassing the notion of good, better, best can be represented equally well by the values $\{1, 2, 3\}$ or by $\{0.5, 1, 10\}$. |
| **Numeric** (Quantitative) | ***Interval*** | $new\_value = a * old\_value + b$, $a$ and $b$ constants. | The Fahrenheit and Celsius temperature scales differ in the location of their zero value and the size of a degree (unit). |
| | ***Ratio*** | $new\_value = a * old\_value$ | Length can be measured in meters or feet. |

The ***dimensionality*** of data (datasets) is the number of features (attributes) that the objects in the data possess [59]. Data with different levels of dimensionality tend to be qualitatively different. High-dimensional data suffers from the curse of dimensionality – the volume of the feature space increases so that the available data becomes sparse [59]. The curse of dimensionality is a problem for some classification and clustering tasks that are based on the density and distance between data points [59].

***Sparse data*** may be cases where, for example, most (asymmetric) attributes have zero value or a value is missing, and low percentages values are non-zero [59].

Different levels of ***resolution*** (e.g. scales in hours, months, or years) reveal different patterns and some patterns lose quality [59]. If the resolution is too fine, a pattern may be buried in noise [59]. In the case of too coarse resolution, the pattern may disappear [59].

Following the resolution of the data, which provides a different view of the data depending on the chosen strategy, we will seamlessly follow ***aggregation levels*** of the data, which can be viewed similarly.

The aggregation levels can be divided into value-level, attribute-level, record-level, table-level, DB-level (see Figure 1.4). From a theoretical point of view, it would be appropriate to extend this list to a higher level, which captures aggregation at the level of multiple databases of one system (system-level) or even at the level of publicly available third-party resources (internet-level). It should be noted that in this context, the aggregation levels are limited to structured data (e.g. tables). Aggregation levels will help us during data quality metrics examination (see Section 1.2.2.1 about Data quality dimensions and metrics) as well.

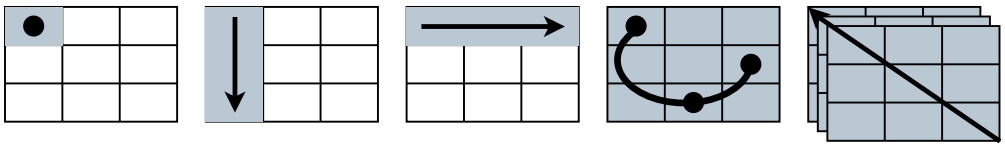

**Figure 1.4:** Illustration of the aggregation levels. From the left side – value-level, attribute-level, record-level, table-level and DB-level.

#### 1.1.2.5 General characteristics of datasets

Basic knowledge of data can be obtained from the dataset type [59]. There are several main variants of datasets that will be presented in this section. One possible grouping of dataset types is as follows:

1. **Record data** (flat files, relational databases (see Section 1.1.2.6 about Diversity of data stores) consists of a collection of records, each of which consists of a fixed set of attributes [59].

    a) ***Transaction data*** consists of transactions where each transaction involves a set of items (e.g. sales orders, invoices, travel records) [59].

    b) ***Data matrix*** has data objects (vectors) in its collection that have a fixed set of numerical attributes (see numerical attribute in Section 1.1.2.3) that represent dimensions of multidimensional space [59].

c) ***Document data*** is represented by a term vector, where each term is a component (attribute) of the vector and the value of each component is the number of times the corresponding term occurs (ignoring order) in the document [59].

2. **Graph data** (see Graph-Based Stores in Section 1.1.2.6) captures relationships among data objects, and the data objects themselves are represented as graphs [59].

   a) ***Data with relationships among objects*** (World Wide Web data) are objects that are mapped to nodes of the graph, while the relationships among objects are captured by the links between objects (e.g. Linked data) and link properties, such as direction and weight [59]. The relationships among objects frequently convey important information (e.g. PageRank) [59].

   b) ***Data with objects that are graphs*** (Molecular structure data) where objects are structured from interconnected subobjects, then such objects are often represented as graphs [59].

3. **Ordered data** (see Section 1.1.2.3 about Data attribute types) has attributes that involve order in time or space [59].

   a) ***Sequential data*** (also known as ***Temporal data***) can be thought of as an extension of record data, where each record attribute can be associated with time (e.g. purchase history of a customer) [59]. Predictive tasks can be performed on sequential data [59].

   b) ***Sequence data*** (e.g. genetic information) consists of a dataset that is an ordered sequence of individual entities, such as a sequence of words or letters [59]. Unlike sequential data, sequence data do not have timestamps [59].

   c) ***Time series data*** (e.g. heights of ocean tides) is a special type of sequential data where each record is a time series (i.e. a series of measurements taken over time) [59]. An important aspect of time data is the (temporal autocorrelation) similarity of time-related objects in some of their attributes [59].

d) ***Spatial data*** (e.g. cross-globe weather data, medical imaging) have some spatial attributes such as positions or areas [59]. An important aspect of spatial data is (spatial autocorrelation) similarity in physically close objects in some of their attributes [59].

In data processing, appropriate attention has to be given to the structure of datasets. One of the processes that focuses on data (record data) structuring is called data tidying. This process creates tidy data. Tidy data[8] have a structure that makes data effortless to analyse and use:

1. each variable forms a column,
2. each observation forms a row,
3. each type of observational unit forms a table [61].

The principles of tidy data are closely tied to those of relational databases and Codd's relational algebra, but with the constraints framed in a language often used by statisticians [61]. Tidy data is focus on a single dataset rather than connected datasets (such in relational databases) [61].

Tidy data (also applies to Record data) can be met in two formats[9] - long (also known as narrow) and wide. The wide format has a single row to describe each element via multiple columns. On the other hand, the long format has multiple rows to describe a single element, where each row holds particular information about the element. The suitability of a given format is chosen according to the type of data analysis. Sometimes it is necessary to convert between these formats, for example, when some data analysis functions require the long format (such as `ggplot2::ggplot()` in R language)[10].

#### 1.1.2.6 Diversity of data stores

An essential part of any technical solution is choosing where to store data. This section will provide an overview of the most important data sources, except relation stores. The relation stores (MySQL, Postgres, Microsoft SQL Server, Oracle Database) only use tables to store and handle data. Relational stores (with transactions at the highest isolation level)[6] are based on ACID

[8] Messy data is any other arrangement of the data [61].
[9] https://kiwidamien.github.io/what-is-tidy-data.html
[10] https://www.datacamp.com/community/tutorials/long-wide-data-R

(Atomicity, Consistency, Isolation, Durability) consistency model. However, striving for full consistency is not always desirable [6]. For example, web-based applications mostly require high availability and the ability to continue working if a computer node or a network connection fails [6]. This has led to such network partition tolerant systems that use replicated computer nodes and a softer consistency requirement called BASE (basically available, soft state, eventually consistent) [6]. This technological solution is also related to CAP theorem, which states that in any massively distributed data management system, only two of the three properties consistency, availability, and partition tolerance can be ensured [6]. According to the CAP theorem, for NoSQL database systems consistency may be fulfilled with a delay to prioritize high availability and partition tolerance [6]. NoSQL technologies are described in the following paragraphs.

***Key-Value Stores*** (e.g. Azure Table, Aerospike, Redis, Riak) is the most simple NoSQL database type (works as a simple hash table) where data are stored as key-value pairs, and the simplest model is easily scalable [6]. Key-Value systems treat data as a single opaque collection, which may have different fields for every record – a data object as a value for another data object as a key [6]. Suitable use cases are, for instance, session data, user profiles, user preferences and shopping carts.

***Column-Family Stores*** (also known as Wide column stores or Column-oriented) (e.g. Google Bigtable, CreateDB, Apache HBase, Cassandra) enhance the key-value concept accordingly by providing additional structure (groups of columns that are often read together). Data objects are addressed with row keys (collection of not necessarily the same columns), and object properties are addressed with column keys (Name-value pair) [6]. Columns in a table are grouped into column families, where data are of the same type, since it is assumed it will be read together [6]. The data objects are versioned with a timestamp [6]. The column-family stores provide high scalability and availability due to key-value pairs, just as with key-value stores [6]. Suitable use cases are, for instance, event logging and content management systems.

***Document-Oriented Stores*** (e.g. MongoDB, CouchBase, Microsoft Azure Cosmos DB, Amazon DynamoDB) are called document-oriented systems that store, retrieve and manage document-oriented information, also known as

semi-structured data (see Section 1.1.2.2 about Types of data) [6]. Documents organized into collections (usually of a similar structure) are identified by a unique key where the value part is a document (XML, YAML, JSON, BSON) [6]. Suitable use cases are, for instance, event logging, content management systems, blogs, web analytics and e-commerce applications.

***Graph-Based Stores*** (e.g. Neo4j, Amazon Neptune, Apache Hama) are similar to document-oriented stores, but they are enhanced with a layer of relations, which allows them to link documents for rapid graph traversal [6]. In graph data stores, data is stored as nodes and edges, respectively, and follow structuring schema belonging to the graph (see Section 1.1.2.5 about General characteristics of datasets). Nodes and relationships contain data in the form of key-value pairs [6]. A suitable use case is for graph structures such as social networks and recommendation engines.

Individual data sources use different types of data formats, according to the requirements of the database system. Unfortunately, no uniform format fulfills all the requirements of data sources, so we must consider heterogeneity in data formats. Some formats are Comma Separated Values (CSV), Tab Separated Value (TSV), Extensible Markup Language (XML), and JavaScript Object Notation (JSON). Some modern data formats also exist for large data storage that are supported by Big Data Frameworks (Hadoop, Apache Spark). These formats include Avro, Parquet, and Apache ORC (Optimized Row Compressed). A list of other formats and possibly their description is beyond the scope of this work.

These different models, technologies, and related formats can be, to some extent, handled by data warehousing systems. A data warehouse system gathers heterogeneous data from several sources (e.g. relational/nonrelational databases, CSV, XML) [30]. It integrates them into a single data store to perform faster analysis and make better decisions via business tools (Business Intelligence, Business Analytic). The main component of the data warehouse are ETL/ELT (extract, transform and load) processes that selects data from the heterogeneous sources, resolves problems in the data, converts it into a common model appropriate for research and analysis, and writes it to the target data warehouse [30]. The disadvantage of data warehousing over traditional

software systems is the difficulty of testing due to heterogeneous sources and voluminous data [30].

A data warehouse is a repository of structured, processed and filtered data for a known schema and purpose [30]. On the other hand, there is a repository focused on a vast amount of raw data (irrespective of the source and its structure [38]) whose purpose does not have to be defined [48]. This data is only transformed when it is ready to be used [48]. Such storage is called a Data lake. Data lake is a good choice for those who want in-depth analysis (e.g. data scientists) whereas Data warehouse is ideal for operational users (e.g. business professionals) [48].

Some companies try to achieve transactional and analytical workloads on data by combining separate technologies (standalone databases, data warehouses or data lakes), which has long been considered good practice to prevent analytical workloads from disrupting operational processing [55]. In such scenarios, data must be moved (ETL, ELT) from Transaction Systems (OLTP) to Operational Systems to Analysis Systems (OLAP), slowing down processing and significantly impeding integration and the ability to gain real-time insight and analytics [55]. In cases where it is necessary to have real-time information from the data (e.g. the emergence of streaming data from sensors, immediate personalized recommendations when placing goods in the shopping cart), this scenario (complex architecture) is insufficient or at least restrictive [55].

Companies responded to this shortcoming by introducing new technology and database implementations (such as SAP HANA, VoltDB, Aerospike, MemSQL, Apache Kudu), due to SSD and RAM memory cost reductions [36] – translytical database (or also HTAP) [36][55]. The general idea of a translytical database (a combination of words ”transaction” and ”analytics” [55]) is to have a single unified technology layer that provides the basis for both application transactions and analytics in real-time without sacrificing transactional integrity, performance, and scale [35]. Copying data from operational databases to data warehouses and data marts for analytical processing not only duplicates the data, but every complex ETL process might introduce data quality problems which leads to inconsistencies in the reporting [36].

#### 1.1.2.7 Data categories

Data categories are groupings of data with common characteristics of features. It is useful to know the different categories of data and their relationships and dependencies, because data for each category should be treated differently, even in the case of data quality. Floridi's [22] first more theoretical categorization of data includes four types of data:

- Primary data – As the name implies, these are the principal data (e.g. a simple array of numbers, or the contents of books in a library) stored in a database store, which is generally designed to be primarily conveyed to the user [22].
- Metadata – These are secondary indications (e.g. location, format, updating, availability, copyright restrictions) about the nature of the primary data [22].
- Operational data – These are related to the operation, use, performance of the data source [22].
- Derivative data – These are data that can be extracted from the above data types [22].

There are no strict boundaries between the different data types above, as it depends on how they are used.

In this section, the more technical categorization of data will be discussed (see Figure 1.5). In this technical case, data can be classified into the following five categories:

- ***Master data*** – Master data are highly valuable key data that describe the principal entities of a given domain that play a crucial role in transactions on that domain [41]. Master data are usually recognized by nouns [45] (e.g. customers, accounts, vehicles, patients, products). It is essential to ensure consistency (e.g. synchronization of their properties) because master data tend to be used by multiple systems [41][45]. Generally, master data are created once, used multiple times, and occasionally changed [41]. Master data are grouped into master records, which may include associated reference data [45]. For example, reference data may be a country record (e.g., Czech Republic, Slovakia) field within an

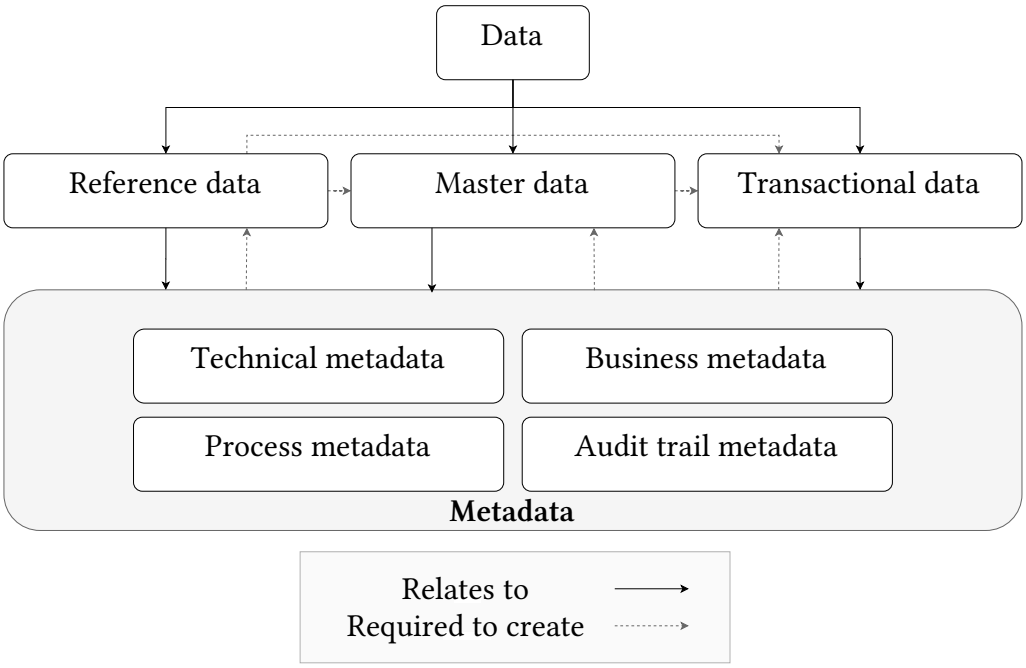


**Figure 1.5:** Data and Metadata categories – inspired by sources [45][41].

address in a customer master record. Errors in master data can have significant cost implications [41]. Such an error could be a wrong address of the customer, causing correspondence, bills or shipments to be sent to the wrong address. Already from this example, it is clear that errors on master data can have significant business impacts (e.g. financial loss or loss of credibility).

- ***Reference data*** – Reference data, as the name implies, are designed to be referenced by other data, such as master data or transactional data, and to provide standard terminology and structure across the organization's systems [41][45]. For example, reference data could be valid value lists, state abbreviations, gender, product types, ZIP codes or HTTP status codes. Standardized reference data are essential for data interoperability and data integration [45]. There are standardization groups (e.g. ISO, CEN) [45] providing, mandating and maintaining reference standards (e.g. ISO 3166-1 with the standard of currency and country codes)[11] to

[11] https://www.iso.org/standard/63545.html

reduce failures caused by inconsistency across organizations. These are also internal reference lists within the organization (e.g. customer and account status codes), to ensure consistency across the organization by standardizing these reference data [41]. Changes in the reference data should be rather rare [41] and thoughtful (such as adding or changing an item name in the reference list). Creating a new master data element sometimes implies the creation or maintaining of reference data [41] (e.g. adding new headphones with a new Bluetooth version to an e-commerce system and extending a reference list with Bluetooth versions).

- ***Transactional data*** – Transaction data describe events at a certain point in time within a domain and represents the largest volume of data in the enterprise [41][45]. In simplicity, there are data from transaction events. Transaction data describes relevant internal and external events in the organization [45] (e.g. sales orders, invoices, purchase orders, credit card payments or shipments). This data are typically grouped into transaction records that include associated master and reference data (see Figure 1.6) [45]. Transaction events are usually associated with a verb (e.g. Create a Sales order) and generate transaction data (e.g. Sales order) [41].

- ***Historical data*** – Transaction data having a time dimension become historical upon completion of the transaction [41]. Historical data at a certain point in time contain significant facts that should not be altered, except for error correction [45][41]. This category of data is essential for security, compliance and forecasting (e.g. financial or stock market forecasts) [45]. One example of historical data could be a change in the customer's last name in the master data, which causes the old master record to become historical data [41].

- ***Metadata*** – Metadata are structured data defining other data [41]. For example, in Figure 1.6, the master data (Product Record) is described by the metadata product information. One of their main tasks is to facilitate data retrieval, interpretation and use [45]. Products can be recommended to customers based on their metadata about browsing e-commerce web sites. Metadata are categorized into the following most common subcategories:

- ***Technical metadata*** is used to describe the technical aspects of data within the storage technologies used (e.g. table column names, length, type, primary or foreign key) [45][41].
- ***Business metadata*** describe non-technical functionality and aspects of data and their use within the business (e.g. column definitions, business terms and rules, key performance indicators (KPIs)) [45][41].
- ***Process metadata*** describe operational information about the operation of systems that generate, maintain, and deliver data (e.g. ETL process logging - start time, end time, CPU seconds used, number of rows read from the target) [41]. In the event of a process failure, the process data are used to solve issues caused by the process failure [41]. Some organizations create business metadata from process metadata, which can be used, for example, to enhance the company's ability to compete [41].
- ***Audit trail metadata*** are a specific type of metadata protected from an alteration of recorded information such as information about capturing how, when, and by whom data were created, accessed, used, updated or deleted [45]. These are metadata for security purposes, compliance checks, or data incident investigations [41]. For security reasons, this type of data is often stored separately from other data [41] or are adequately protected from access by unauthorized data storage users.

The book [45] provides historical and temporary data as additional data categories. Temporary data are usually stored in memory for speed data access and is primarily intended for technical purposes (e.g. copy of a table that is created during a processing session to speed up lookups) [45].

Figure 1.5 shows the associations between different data categories. Knowledge of these associations is essential for understanding the transmission of data quality issues and the interconnectivity of these categories [45]. Some reference data are required to create master data, and master data are required to create transaction data. Sometimes, the reference data are directly related to the transaction data, without relating to the master data. Metadata serves for

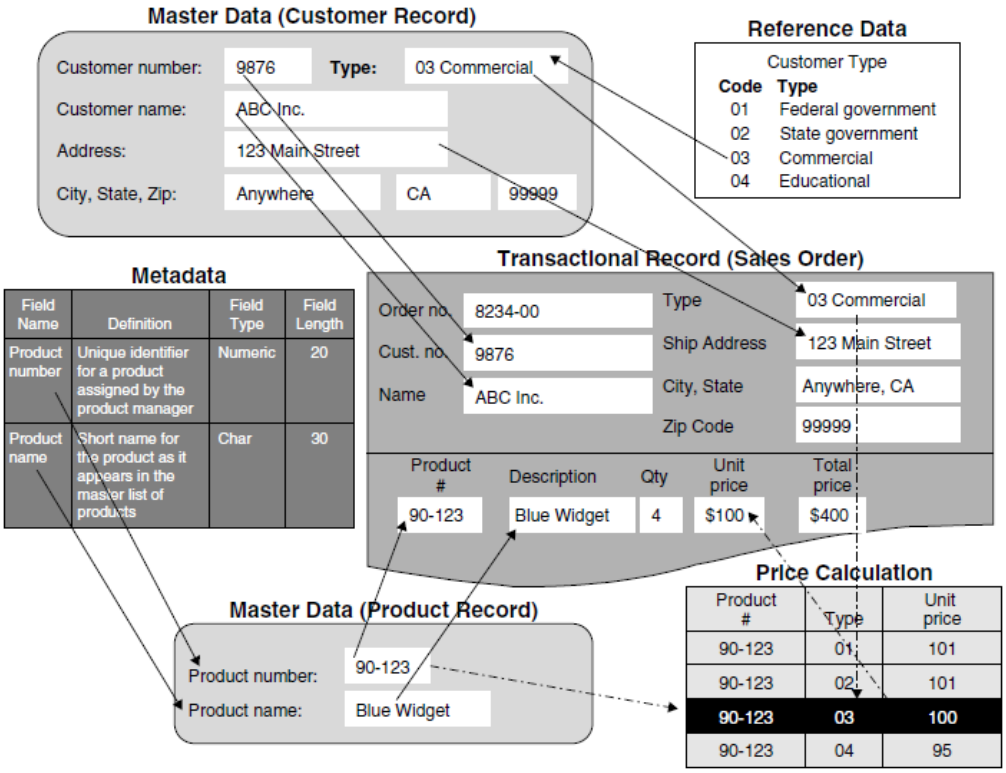


**Figure 1.6:** An example of data categories [45].

information extension of other categories. From a historical data point of view, it is sometimes necessary to preserve related data from other categories. Otherwise, the information context may be lost. Described data categories have different properties and meaning, thus it is important to think about this data categorization when solving data quality [45].

### 1.1.3 Quality consequences of Data life flow

This section describes the consequences of changes in the data and puts them into the context of data quality to follow up on the next sections about data quality (1.2) smoothly.

Data changes can come with processes or methodologies that usually aim to find knowledge from the data. Among the known processes/methodologies, for example, we can mention the KDD process (Knowledge Discovery in Databases), which tries to find useful information or patterns in the data [21]. But before we get to the application of a method (in this case a data

mining method) the data will go through selection, preprocessing, reduction or transformation.

These changes are relatively significant and often require expert interaction. For example, data preprocessing interferes with data considerably. It is subject to expert interaction and is difficult to automate due to its complexity. This type of complex data transformation is prone to data issues, and information quality is instead transferred to human precision and responsibility.

By moving to a higher abstraction, to the life cycle of information (see Figure 1.7), the information (data) can be analyzed in its life flow. We also encounter these cycles in several forms [37], but still sticking to subjectivity and non-universality in the data.

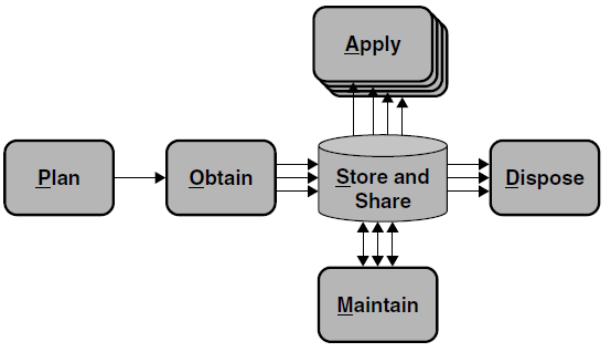


**Figure 1.7:** The Information life cycle is not a linear process [45].

In this data life cycle, we understand information as a resource that must be appropriately managed during its life cycle in order to reach its potential [45]. We can observe that this is a non-linear and iterative process which includes the following phases (used information source [45]):

- ***Plan***. Preparation for the source. For example, objectives identification, information architecture and design planning, development of standards and definitions.
- ***Obtain***. Acquire the resource. For example, loading or creating a resource.
- ***Store and Share***. Information holding and distribution. For example, storing data in a database and sharing it over a network.

- ***Apply***. Use resources to achieve goals. These are all ways of using information. For example, creating a report for the management or running automated processes.
- ***Maintain***. Ensure that the resource is working properly. For example, updating, changing or enrichment of data.
- ***Dispose***. Cancel a resource when it is no longer in use. For example, archiving or deleting records.

Each of the above phases is related to value, price and quality [45]. The company benefits from the value of the resource if the cost of using the resource is lower than the price of the value obtained. The information must not only be correct, but also useful and appropriate quality.

Data quality is related to each activity in each phase of the life cycle [45]. We got another abstraction above – the quality of individual activities on the data, which to my best knowledge is no longer specified.

Quality is a critical part of the information. If the wrong information is applied repeatedly, negative consequences occur [45]. For example, costs or loss of credibility of the company. If the correct information is applied more than once, the value of the information itself will increase [45].

Data is very prone to the ”butterfly effect” [41]. This concept comes from the science of chaos theory and is defined as a sensitive dependence on initial conditions, in which a small change in one state of a deterministic non-linear system can lead to large differences in a later state [41].

In the case of data, unexpressive information error can be exponentially propagated across the system in a significant manner. For example, there may be a situation where time data transmitted between systems is converted from UTC to the local zone due to an upgrade of the support library version[12].

These may be time errors in the order of hours. This time error can be used in an upstream system to calculate the time range, and this erroneous range can be further used in several other systems. For example, it may be a machine learning model that erroneously generates information for a business

[12]https://github.com/Azure/azure-storage-net/issues/634)

report. Tracing the error from report to error in the library upgrade requires considerable expert effort.

It would be adequate to avoid these ripple effects in the data. If we want to monitor data activity to avoid ripple effects, then is appreciated to automate data quality control with a focus on the initial stages of data flow to avoid finding errors in an exponentially larger data flow search space.

## 1.2 Understanding data quality

Data are often far from perfect (see Section 1.1 about Understanding data). Therefore, improving data quality is a way how to get data closer to perfection. Before focusing on concepts of data quality measurement (see Section 1.2.2 about Data quality measurement), it is necessary to determine when data are considered to be quality (see Section 1.2.1 about Data quality in theoretical context). Data quality is the extent to which the data meet the requirements and purpose for their use and trustworthiness [45][41].

Another way to understand the concept of data quality measurement is to divide it into categories, understand their relationships and quality purposes. In simplicity – accuracy, completeness, timeliness and consistency can be included among the elementary attributes of data quality. These data quality attributes, are called the data quality dimension (see Section 1.2.2.1 about Data quality dimensions and metrics). Each of these dimensions describes the possible requirements coming from quality issues in the data and the metrics that try to measure and control the data quality.(e.g. Number of empty values) [45]. Metrics can also determine the success of data quality methods.

The data quality tools (see Section 1.3 about Existing data quality tools review) commonly include a range of critical data quality methods, such as data profiling, data quality measurement and monitoring and data cleansing which present the core categorization of data quality activities (see Section 1.2.1 about Data quality in theoretical context).

The following sections build on the previous section that describes the data itself (see Section 1.1 about Understanding data). This section is intended to help understand data quality measurement from a general perspective.

### 1.2.1 Data quality in theoretical context

When talking about data quality, the key starting point is to find out what is meant by the term. The term Quality Data is frequently described in the literature as ”fitness for use” [20] [19] and captures the high subjectivity and contextual dependence of the term [19]. Due to the subjectivity of the term data quality, there is no agreement on the definition of the term [20].

Data and information quality are often used interchangeably in data quality literature [19] [45], although both terms can be distinguished (see Section 1.1.1 about Data in theoretical context). The following definition of Information Quality applies to both data and information:

”Information quality is the degree to which information and data can be a trusted source for any and/or all required uses. It is having the right set of correct information, at the right time, in the right place, for the right people to use to make decisions, to run the business, to serve customers, and to achieve company goals.” [45]

From this point, we will stick to the term data quality because the work focuses on processing objectively, automatically retrievable facts.

In relation to the section 1.1.1 (Data in theoretical context), data quality dimensions can be categorized according to semiotic levels, such as in Table 1.3.

The division of dimensions according to semiotics levels is too granular, theoretical and difficult to grasp for industrial use in practice. The better technical nature of the data and customers needs are captured in the conceptual framework of data quality by Wang and Strong [60], which organizes the dimensions into four main areas - intrinsic, contextual, representation and accessibility data quality.

***Intrinsic data quality*** (e.g. accuracy, objectivity, believability and reputation) is that data has quality in itself [60]. ***Contextual data quality*** (e.g. relevancy, timeliness, an appropriate amount of data, completeness) is that data quality must be considered within the context [60]. ***Representation data quality*** (e.g. interpretability, consistency, ease of understanding) includes aspects related to data format (concise and consistent representation) and meaning of data (interpretability and ease of understanding) [60]. ***Accessibility data quality***

**Table 1.3:** Semiotics levels related to DQ dimensions, adapted from [31].

| Semiotics levels | DQ dimensions | Example |
|---|---|---|
| Empirics | Accessibility | Sequence records are easily and quickly retrievable for access. |
| | Timeliness | Sequence records are sufficiently up-to-date. |
| Syntactics | Accuracy | Sequence records are correct and free of error. |
| Semantics | Believability | Sequence records are regarded as credible and believable. |
| Pragmatics | Completeness | Annotated sequence records are not missing and are fully annotated. |

means that the system must be accessible, but secure at the same time [60]. In summary, this means that high-quality data should be intrinsically good, contextually appropriate for the task, clearly represented and accessible to the data consumer [60].

The core data quality activities can be divided as follows [19]:

- ***Data profiling*** – the process of analyzing a dataset to collect metadata and is an essential task prior to any data quality monitoring or monitoring activity to get insight into a given dataset (e.g. the number of distinct or missing values in the column) [19].

- ***Data quality measurement*** – the process of evaluating data quality within the dimensions of data quality (e.g. identification of root causes, necessary improvements and data corrections) (see Section 1.2.2 about Data quality measurement) [45].

- ***Data cleansing*** – describes the process of correcting erroneous data or data issues (e.g. data standardization, de-duplication, and matching) [19].
- ***Data quality monitoring*** – the process of ongoing measurement of data quality (the term is mainly used implicitly without an established definition and common understanding in literature) [19].

Another theoretical division of data quality is outside the scope of this work. The data quality organization above should provide the reader with an insight into the data quality. Furthermore (see Section 1.2.2.1 about Data quality dimensions and metrics), the dimensions will be discussed without reference to their categorization.

### 1.2.2 Data quality measurement

Measuring data quality is a big challenge, as the answer to the question "How to measure/assess data quality?" is not unequivocally answered [19].

Assessment is often used as a synonym for measurement, but in the data quality literature we can find the difference between these terms [19]:

"Assessment is an evaluation or estimation of the nature, ability, or quality to something and extends the concept of measurement by evaluating the measurement results and drawing a conclusion about the object of assessment." [19]

This work uses the term measurement of data quality because one of the broader goals of the work is the automation of data quality, which seeks to achieve greater independence from the interpretation of results by the user.

In this work, the term data quality measurement is also used in the context of various data quality activities that are understood as an auxiliary part of the measurement. Data profiling activity can help to gain essential insight into the data. Data quality monitoring is understood here as ongoing measurement. The work will not deal with tasks for automatic data cleaning. Methods for cleaning (e.g. de-duplication) could be used only as a basis for detecting data quality problems. The determination of the theoretical boundaries of data quality helps to design experiments for data quality (see Section 3 about Experiments and Results).

Data quality measurement is closely related to data quality dimensions and assigned metrics (see Section 1.2.2.1 about Data quality dimensions and metrics) [45]. A partial answer to the question at the beginning of this section is to use useful data quality metrics to measure data quality. According to an article [50], most data quality measurements are performed on an ad hoc basis to solve a specific problem, and the fundamental principles necessary to develop applicable metrics in practice are lacking. This assumption will be discussed later (see Section 1.3 about Existing data quality tools review) on contemporary data quality tools.

Data quality measurement (assessment) can be subjective ("soft dimensions"[19] – e.g. domain-specific business rules) or objective ("hard dimensions"[19] – e.g. accuracy, completeness, timeless or general integrity rules (a birth date cannot be in the future)) [50]. Subjective measurement of data quality reflects the needs and experiences of stakeholders [50].

Objective measurement of data quality can be task-independent or task-dependent [50]. Task-independent metrics are able to assess the state of any data without knowledge of the application context [50]. On the contrary, task-dependent metrics require application context knowledge (e.g., organization's business rules) [50]. This work focuses on the objective measurement of data quality, as it is potentially easier to automate.

#### 1.2.2.1 Data quality dimensions and metrics

Data quality is based on a multi-dimensional concept [19], where aspects of data quality are described by data quality dimensions for which one or more metrics can be used as quality indicators [45].

Data quality science works try to define a list [50][45] of data quality dimensions. This list typically includes dimensions such as ***Accuracy***, ***Completeness***, ***Timeliness***, and ***Consistency*** [19]. The dimensions often overlap, are vaguely defined, are ambiguous and are not justified adequately in theory [57]. For these reasons, the work will only deal with the dimensions listed above and clearly defined metrics.

It should be noted that it is often not clear how to map dimensions and metrics to a practical implementation [19].

A metric is a function that maps a data quality dimension to a numeric value that allows the dimension fulfilment to be interpreted [19].

Following the section dealing with aggregation levels (see Section 1.1.2.4 about General data characteristics), it is worth mentioning that data quality metrics can be measured at different aggregation levels (e.g. weighted arithmetic mean of the calculated metric result from the previous level)(see Figure 1.4).

The following sections describe the key dimensions of data quality and essential metrics for numerically interpreting a given dimension. This is not an exhaustive list of metrics, as their purpose is to give a complete picture of how metrics can be calculated. A more detailed discussion of data quality metrics is in the literature [50][19][45][9]. The specific metrics used in current data quality tools are analyzed in [19].

#### 1.2.2.2 Accuracy

Accuracy (sometimes described as the most important data quality dimension [19]) could be defined as the closeness between a value $v$ and a value $u$, considered as the correct representation of the real-life phenomenon that $v$ aims to represent and value of $u$ as its incorrect representation [9]. In practice, it is essential to define rules when a data unit is considered to be an error [50]. Accuracy is related to the free-of-error rating metric, which is calculated as the number of erroneous data units divided by the total number of data units subtracted from 1 [50][19]:

$$\texttt{Free-of-error rating} = 1 - \frac{\texttt{Number of data units in error}}{\texttt{Total number of data units}} \tag{1.1}$$

This simple ratio adheres to the convention that 1 represent the most desirable and 0 the least desirable score [50].

#### 1.2.2.3 Completeness

Completeness can be very generically defined as "the extent to which data are of sufficient breadth, depth, and scope for the task at hand" [9]. Three types of completeness are identified [9]:

1. ***Schema completeness*** is defined as the degree to which entities and attributes are not missing from the schema [9][50].

2. ***Column completeness*** is defined as a measure of the missing values for a specific column in a table [9][50].

3. ***Population completeness*** evaluates missing values with respect to a reference population (e.g. a column should contain at least one occurrence of all 20 states, but it only contains 17 states) [9][50].

Each of the tree types can be measured by taking the ratio of the number of incomplete items to the total number of items and subtracting from 1 [50]:

$$\texttt{Completeness} = 1 - \frac{\texttt{Number of incomplete elements}}{\texttt{Total number of elements}} \tag{1.2}$$

#### 1.2.2.4 Timeliness

Timeliness describes how current the data is for the task at hand [9]. It is possible to have up-to-date data that is actually useless because it is too late for a specific usage [9]. Timeliness is closely related to the notions of currency (update frequency of data) and volatility (how fast data becomes irrelevant) [19]. Timeliness can be calculated as follows [19]:

$$\mathtt{Q}^{\omega}_{\mathtt{Time}}(t) \coloneqq exp(-\boldsymbol{decline}(A) \cdot t), \tag{1.3}$$

where $\omega$ is the considered attribute value and $\boldsymbol{decline}(A)$ is the decline rate, which specifies the average number of attributes that becomes outdated within the time period $t$ [19].

#### 1.2.2.5 Consistency

According to Batini and Scannapieco [9], "the consistency dimension captures the violation of semantic rules defined over (a set of) data items, where items can be tuples of relational tables or records in a file" [9]. Consistency can be calculated similarly to the previous dimensions [50]:

$$\texttt{Consistency} = 1 - \frac{\texttt{Number of violations of a specific consistency type}}{\texttt{Total number of consistency checks}} \tag{1.4}$$

## 1.3 Existing data quality tools review

Data quality tools are a vital data measure for the benefit of the business and should serve legitimate business needs [45]. Due to urgent business requirements, the complexity of data (see Section 1.1.2 about Heterogeneity in data) and its storage space is increasing, putting pressure on efforts to care for their quality [17].

New features are emerging like automation, machine learning, business-centric workflows and cloud deployment models due to the strain on the market with data quality tools [17]. According to a survey [19] from 2019, 667 software tools dedicated to ”data quality” were identified. The primary areas of these tools include data cleansing, data integration, master data management, and metadata management [25]. Based on demand, organizations prioritize the development of areas such as audience, governance, data diversity and latency [17]. On the other hand, data quality tool vendors focus on enhancing areas such as analytics, intelligence, deployment and pricing [17]. From the latest Gartner data available, the data quality tools market grew 11.6% in the year 2017 [17].

In lower abstraction, data quality tools eliminate data issues such as formatting errors, redundancy, inconsistency or wrong data types. They enable data errors and anomalies to be detected using algorithms and technologies (e.g. lookup tables) supporting automation of data quality processes [25]. The use of rules, automated processes and their detailed logging is considered as essential for supporting the data quality of an organization [25]. Current challenges for data quality tools include data consolidation associated with ETL tools, data validation reconciliation, data analytics, and managing large amounts of heterogeneous data [25]. One of the priorities of today’s data quality tools is also to facilitate the work of data stewards [17].

The survey of data quality measurement and monitoring tools found the following findings based on measurements of 667 data quality tools in 2019:

- 50.82% of data quality tools examined were domain-specific (dependence on data type or proprietary tool),
- 16.67% of data quality tools focused on data cleansing without proper data quality measurement strategy,

- data profiling (to some extent) is the majority supported data quality technique with the potential extension to multi-column profiling and dependency discovery,
- no tools were found that implement a wide range of data quality metrics for the most important data quality dimensions (see Section 1.2.2.1 about Data quality dimensions and metrics),
- some data metrics implementations had implementation errors, some were applied to the attribute-level only (without aggregation), or required a potentially non-existent gold standard,
- more versatile data quality tools provide data quality monitoring as a paid premium feature,
- there are exceptions to open-source data quality monitoring tools, such as Apache Griffin or MobyDQ, which support rule automation but lack predefined functions and data profiling capabilities [19].

Users demand greater versatility of tools in terms of wider use of data management and information management capabilities [17]. Therefore, a closer interaction of data quality tools with data integration tools, information governance frameworks and master data management (MDM) products is appreciated [17]. Moving data quality tools to the cloud (SaaS for data quality) is becoming a modern trend [17]. It is important to realize that data quality tools can be useless and ineffective if a customer does not know what to want from such tools, what to expect from them and if the customer does not consider important prerequisite information such as how and where the organization store data, data flow, data usage and pricing models as well. Nowadays, it is not just a matter of the IT industry. It requires cooperation with executives and users [17]. Thus, people with the appropriate knowledge of business needs, processes, and data work should cooperate closely with those who have the skills to implement and use the technology [45]. It is also necessary to keep in mind the scenarios, which use multiple tools for solving data quality in the organization. Purchasing a data quality tool without knowing why and how a customer will use it – the customer may get poor data quality or a low price-performance ratio of the data quality tool.

### 1.3.1 Selection criteria for data quality tool

In the previous section, it has been emphasized several times that it is important to first understand the needs of data quality and then to find a suitable tool. This section discusses the criteria that play an essential role in the selection of a data quality tool. Knowledge of these criteria will help with the selection of relevant methods for research in the following section (see Section 2 about Data quality measurement methods). First we look at the data quality criteria from a higher abstraction perspective and then move on to more detailed criteria.

General data quality needs can be divided into three categories:

- ***Data understanding*** – being aware of and comprehending the meaning, content, location and behaviour of data [45].
- ***Data repairing*** – correcting or updating data (closely related to phase Data Preparation in CRISP-DM process) [45].
- ***Data monitoring*** – checking data proactively by applying tests (such as quality control rules) that draw attention to anomalies in data to prevent business damage [45].

Another view of selecting a data quality tool is as follows:

- ***Identify your data challenges*** – Identify and analyze the data field of your organization. It is essential to analyze existing data sources, current tools in use, and data quality [25].
- ***Understand what data quality tools can and cannot do*** – There is no data quality tool to repair completely broken, incomplete or missing data [25]. It is advisable to keep such limits in mind. The concept of GIGO (Garbage in, Garbage Out) also applies here, if the data framework of the organization suffers from fundamental deficiencies, these deficiencies are transferred to the tools of data quality and their potential will not be fully exploited.
- ***Understand the strengths and weaknesses of various data cleansing tools*** – Not all data quality management tools are equivalent [25]. Some are application-specific, such as Salesforce or SAP, others are focused on finding data quality issues in physical mailing addresses or e-mails, some

dealing with IoT data [25]. It is necessary to understand the technical capabilities of the data cleansing tool (e.g. features, level of automation), the level of security and the pricing model of the tool [25].

Gartner publishes a more detailed list of key capabilities that organizations should consider when choosing a data quality tool:

- ***Connectivity*** – Capability to access, and apply data quality rules to a wide range of data (see Section 1.1.2 about Heterogeneity in data) and data sources (see Section 1.1.2.6 about Diversity of data stores) [17].
- ***Data profiling, measurement and visualization*** – Capabilities to understand and analyse characteristics of varied types of data (see Section 1.1.2.2 about Types of data) and data issues to support data quality [17].
- ***Monitoring*** – Capabilities to assist users continuously respond to data quality issues, monitor data and assurance data quality [17].
- ***Parsing*** – Capabilities to decompose data into its component parts based on the need to better understand the internal characteristics of data [17].
- ***Standardization and cleansing*** – Capabilities to standardize (e.g. industry or local standards, business rules, knowledge bases) and cleanse the data to achieve specific formats, values and layouts [17].
- ***Matching, linking and merging*** – Capabilities for matching, linking and merging related data entries within or across datasets via a variety of techniques, such as rules, algorithms, metadata and machine learning [17].
- ***Multidomain support*** – The capability to support a specific area of data such as customer, product, property and location to ensure sufficient quality [17].
- ***Address validation/geocoding and other validation*** – Support for location-related data standardization and cleansing [17].
- ***Data curation and enrichment*** – The capability to integrate externally sourced third-party data to improve completeness and add value (such as Demographic Data Enrichment, Geographic Data Enrichment)[17].

- ***Issue resolution and workflow*** – Capabilities to correctly set the process flow and user interface that enables business users to identify, quarantine, assign, escalate and resolve data quality issues [17].

- ***Metadata management*** – The capability to capture, reconcile and interoperate metadata relating to the data quality process to promote greater confidence in data [17]. Indication of how and when the data got to the system and what business rules were applied is important.

- ***DevOps environment*** – Capabilities that facilitate configuration and application of data quality operations [17]. Such as capabilities of Data DevOps that can give increased speed, quality, security, and a great degree of productivity [58].

- ***Deployment environment*** – The capability to deploy data quality operations to the organization's hardware and software [17].

- ***Architecture and integration*** – The capability of commonality, consistency and interoperability among various components of data quality tools environment and third-party tools [17].

- ***Usability*** – The capability to maximize the usability of a data quality tool to the benefit of an organization's business [17].

Based on the above criteria, which should be taken into account when selecting a data quality tool, we can conclude that the effort and time investment in analysing the implementation of a data quality environment into the organization is essential.

### 1.3.2 Data quality tools analysis

Healthy competition in the market for data quality tools shows their importance. There are a plethora of these tools, so a fundamental analysis of existing data quality tools (see Appendix files list – `DQ-tools-analysis.xlsx`) was carried out in the context of the previous findings. Specifically, the analysis is a search of 21 data quality tools and their parameters (such as focus, key features, documentation, support, pricing model, open/close source code, strengths and cautions).

The aim was to obtain basic information about the approach to data quality of these tools, so that the selection of methods for analysis in the following sections is as close as possible to the methods used in practice. Learning from the tools shortcomings is one of the key priorities of this analysis as well.

Information was drawn from publicly available information on the web, studies and articles. The practical analysis was omitted as it is a time-consuming activity outside the scope of this work. A valuable study for analysis was "Magic Quadrant for Data Quality Tools 2019" [17] from Gartner Inc., which includes vendors offering commercial software tools or cloud-based services with data quality features (profiling, parsing, standardization/cleaning, matching, monitoring). The study is focused on the data quality tools market than on functionality itself. Therefore, a survey [19] of 13 free/trial and domain-independent data quality software tools was included in the analysis.

The main findings of the analysis are as follows. Due to the high degree of diversity and complexity in the data, the existence of a unitary and universal tool is doubtful. Therefore, vendors limit data quality tools to specific domains (e.g. finance), technologies (e.g. SAP), data types (e.g. structured data) and data processing (e.g. batch processing) to reduce the complexity of systems.

It seems that maintaining universality is proportional to the cost. This phenomenon can be seen with data quality tools such as SAP, SAS and IBM, which are leaders in the industry [17]. These tools cover a wide range of data quality methods, technologies and support functions, but such scope negatively affects the cost of these tools. Over the years, the complexity and interconnectivity of these tools have increased. It leads to the tendency of vendor-locking [17]. This lock may not only be a purely business intent, but also a limited ability to integrate a complex system to third-party tools. Thus, vendors create a single data quality environment. By purchasing one tool, the customer is pushed in this vendor environment to stay. The environment of vendors is, in some cases, reminiscent of the financial or banking sector, which can be harmful and distract vendors' attention in the wrong technical direction.

Vendors know about the complexity of the data quality systems. Therefore, efforts are being made to move data quality tools to a single cloud environment

(e.g. by combining data integration, data preparation, and stewardship) to be served by all data quality actors in one place [44]. Data quality intervenes in various roles that need close cooperation. Such tools include Talend Data Management Platform[13] or SAS Viya[14]. This step is mainly due to the growth of cloud data storage industry and an increase in demand for cloud integration solutions (e.g. Talend Cloud Integration Platform[15]). Data quality tools are closely linked to these data integration tools and should be given due consideration [44]. If the data progresses to higher levels of the system, the probability of errors or deficiencies in the data increases, so practising for an active approach to data quality is suitable prevention. This preventive behaviour allows to check and measure the level of quality before it even really gets into the core systems [44].

Nowadays, the ability to service data quality of IoT and mobile devices is essential to data quality tools [49]. It is a challenge of unifying and consolidating a wide range of data formats from multiple data streams and control their quality (”real-time data quality”) [49]. In terms of innovation, the use of artificial intelligence and machine learning technologies to simplify and automate processes (automated project configuration, automated metadata discovery, anomaly detection, evaluation of results, etc.) is an essential approach of the data quality vendors leaders [17].

Users consider important to have professional technical support [17]. It should be noted that the support is closely related to the price of the data quality tool. For this reason, it is worth investing in the documentation, simplifying the process of deploying and integrating the tool, supporting the community, and the ease of use of the tool.

From the analyzed tools, three ”providers” were selected that stand out – Talend[16], Informatica[17] and MobyDQ[18].

[13] https://www.talend.com/products/data-integration/data-management-platform/
[14] https://www.sas.com/en_us/software/viya.html
[15] https://www.talend.com/products/integration-cloud/
[16] https://www.talend.com/
[17] https://www.informatica.com/
[18] https://github.com/ubisoftinc/mobydq

Data quality tools characterized by ease of use and an active open-source user community include Talend Open Studio (TOS) for Data Quality (and a free open source license) and Talend Data Management Platform (a user-based subscription)[19]. The Talend has doubled the number of customers with data quality licenses between 2017 and 2018 [17]. It is a relatively young company (founded 2005) with a higher degree of adaptability. On the other hand, the tool is criticized for the lack of technical support, the difficulty of upgrading and migrating data between versions, and the lack of monitoring, reporting and scheduling features for data quality processes [17]. Talend Open Studio for Data Quality is the leading open-source data profiling tool [19].

In contrast, Informatica has been on the market since 1993 and is considered the ”gold standard” among data quality tool providers [17]. Some users considered that its prices are high and the licensing models are complicated, but the company responded quickly and introduced more dynamic models such as pay-as-you-go [17]. It should be noted that it is a sophisticated tool, and it is necessary to consider the financial return of such investment when buying it. Informatica is one of the most versatile and innovative tools – introduces innovations mainly in the field of machine learning such as metadata-driven machine learning to identify data domain consistency, outliers and errors, and range of tasks related to Internet of Things (IoT), MDM, data governance and content-driven analytics [17]. It has a robust global ecosystem, with over 500 partners for all of its products, including Accenture, Amazon, Cognizant, Deloitte, Google and Microsoft [17]. On the other hand, users report that the initial setup is complex, time consuming to teach users to use tools, poor stability (restart the server at least once a month), and outdated visualization capabilities [17].

A new categorization of data quality tools was observed from the analysis. The above-mentioned tools (mostly general-purpose tools) are, in some ways, aimed at less technical users, which can bring unsolicited complexity into its implementation. If we focus primarily on data engineering teams that are expected to have the higher technical knowledge, we can avoid complex solutions due to omitting user features. And bring an active approach to data quality closer to the data source [44]. This avoids the likelihood that poor quality data will bubble through the system to higher layers where it can

potentially cause more serious damage. Detecting the source of errors in data within a long data footprint is costly and time-consuming. These types of tools are Apache Griffin[19] and MobyDQ. Let's take a closer look at MobyDQ.

"MobyDQ is a tool for data engineering teams to automate data quality checks on their data pipeline, capture data quality issues and trigger alerts in case of anomaly, regardless of the data sources they use." [53] It is an open source project inspired by an internal data quality project developed by Ubisoft Entertainment [53]. In contrast to Apache Griffin, MobyDQ could be installed quickly and straightforward, based on detailed documentation provided on GitHub [19]. On the other hand, MobyDQ does not provide any data profiling functionality [53]. A tool of this type offering a wide range of ML-based methods to support automation may present a potential void in the data quality tools market.

[19] https://github.com/apache/griffin

Chapter 2

# Data quality measurement methods

This section assesses methods for measuring data quality. The main goal is not only to theoretically describe the methods in detail, but to assess their capabilities, limits and find their practical use for measuring data quality.

In the first section is introduced the analysis of state of the art approaches to enhance data quality in the context of the measurement (see Section 2.1 about Analysis of state-of-the-art approaches to enhance DQ).

The next section (see Section 2.2 about Data quality measurement using Autoencoder) follows up on the previous section, with the difference that the AI approaches in it (i.e. Autoencoder, Association Rule Mining) are analyzed in more detail and described theoretically. The analysis of the approaches will also serve as a theoretical basis for experiments (see Section 3 about Experiments and Results).

## 2.1 Analysis of state-of-the-art approaches to enhance DQ

This section summarize interesting AI and non-AI data quality methods. Trivial methods of measuring data quality will be omitted, such as measuring cardinality (e.g. number of rows, empty/null values) or value distribution (e.g. min, max, mean, quartiles, cardinality).

Current general-purpose data quality tools do not fully utilize the potential of the ML-based method [19]. Although several vendors claim to implement ML-based methods, the survey [19] found that no or only limited documentation of concrete ML-based algorithms is available [19].

According to the survey [19], it is important not only to focus on enhancement of the detection of data quality errors via AI/ML methods but also on the desirable core characteristics (such as wide applicability of the methods, easy to use, interpret, store and deploy, and the methods should have short response times) [19].

The following approaches were evaluated as interesting methods with the potential to measure data quality:

- ***Clustering*** – Clustering approach can be either used to detect outliers in a single column, or to detect similar or duplicate record within a table [19]. In the OpenRefine data quality tool, the clustering approach[20] is used to find different values that can represent the same thing. As part of the measurement, users could be informed about clusters and their characteristics representing these similar records.
- ***Benford's law***[21] – Benford's law is focus on distribution of first digit in numeric values [19]. A common application is fraud detection (bank transactions) [19]. However, the method could be used to detect anomalies in Machine data (see Section 1.1.2.1 about Data areas) as well. This prevention can, for example, save industrial machines from destruction based on abnormalities caused by a technical defect. The research [19] shows that this method is not sufficiently used in data quality tools.
- ***Semantic data types detection*** – For example, generic semantic data types are codes, date, URLs or identifiers (defined by generic patterns) and non-generic semantic data types include city, country, or name [19]. This approach can have a significant impact on the automation of ML-based methods. Some machine learning methods (e.g. Autoencoder) could benefit from knowledge about the semantic data type of their

[20] https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth
[21] https://en.wikipedia.org/wiki/Benford%27s_law

input values. In an example, Date data type was detected for a feature, and it would then be appropriately tokenized to preserve semantic information. Current implementations rely on dictionary lookups and regular expression matching [32]. However, these implementations have theirs limits (e.g. dirty data and detection of a limited number of data types) [32], but exist a state of the art effort (Sherlock: A Deep Learning Approach to Semantic Data Type Detection developed by MIT [32]) based on a multi-input deep neural network that seeks to overcome these limits.

- ***Automatically Generating Regular Expressions via Genetic Programming*** – In the field of the automatic generation of regular expressions exists a research [8] (with an implementation[22] of a web application[23]) based on Genetic Programming achieving very good results. The implementation of Automatic regular expression synthesis is able to work only on examples of the desired behavior [8]. This approach can address context-dependent extractions or widely different formats [8]. The synthesis of regular expressions has the potential to measure data quality in the field of automatic measurement of a number of wrong format records.

- ***Running statistics of streaming data*** – Running statistics (such as variance, mean or standard deviation)[24] could be used to measure data quality on stream data in order to perform continuous online monitoring of outliers. For example, algorithms of this type include Welford's online algorithm[25].

- ***Duplicate detection via NLP approaches with XGBoost algorithm*** – One of the most interesting approaches to detecting text duplicates may be the development of a machine learning model (e.g. XGBoost[26]) with NLP approaches. NLP approaches that would extract features from the text for use in machine learning modeling using XGBoost algorithm are Word2Vec, Bag-of-Words, TF-IDF or Fuzzy string matching[27] (via

[22] https://github.com/MaLeLabTs/RegexGenerator
[23] http://regex.inginf.units.it/
[24] `https://www.johndcook.com/blog/standard_deviation/`
[25] `https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance`
[26] `https://xgboost.readthedocs.io/en/latest/`
[27] `https://github.com/seatgeek/fuzzywuzzy`

Levenshtein Distance). There are prototypes of implementations[28][29] that achieve good results on the Quora dataset[30] for finding duplicate questions.

## 2.2 Data quality measurement using Autoencoder

The section will discuss theoretically the use of neural autoencoders for measuring data quality. A description of the implementation of this approach can be found in its experimental part (see Section 3.3.1 about Experiment 1 – Autoencoder). The neural autoencoder is unsupervised ML technique and feed-forward neural network [51]. The general idea consists of the setting an encoder and a decoder as neural networks that learn to encode the most representative features from input data to the latent space using an iterative optimization process evaluated by a reconstruction error [52][51].

In addition to the encoder and decoder, the autoencoder architecture also consists of a latent layer. The input layer of the encoder and the output layer of the decoder has the same number of neurons. The last layer of the encoder is the already mentioned latent layer. In the case of undercomplete autoencoder (see Figure 2.1), the latent layer has a smaller dimension than the input. The number of neurons in the encoder layers gradually decreases forward, while in the decoder the number of neurons in the layers gradually increases forward. The entire autoencoder could be seen as layers of interconnected neurons, where encoder encodes input data, and then the main goal of the decoder is to reconstruct the input using the latent layer [51].

The primary domain of autoencoders is the ability to discover low-dimensional representations of high-dimensional data (i.e. dimensionality reduction), while still attempting to preserve the fundamental attributes present therein, without explicitly relying on human-engineered assumptions [51]. Based on this elementary autoencoder feature, the autoencoder was selected as an ML method capable of automating data quality measurements.

[28] https://github.com/susanli2016/NLP-with-Python/blob/master/Word2vec_xgboost.ipynb

[29] https://github.com/abhishekkrthakur/is_that_a_duplicate_quora_question

[30] https://www.kaggle.com/sambit7/first-quora-dataset

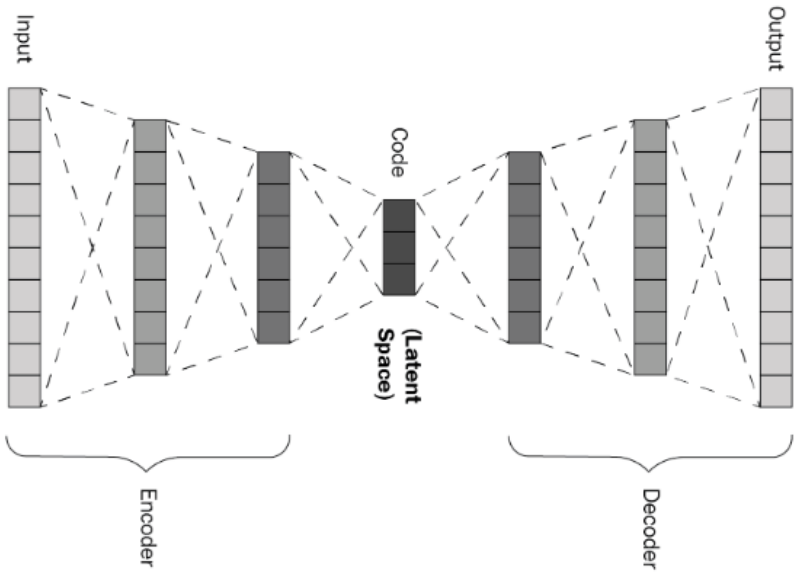


**Figure 2.1:** Example of an undercomplete autoencoder [51].

Before introducing autoencoder in the context of measuring data quality, it is appropriate to get acquainted with the known limits of autoencoders.

- ***Data-specific*** – The usefulness of the autoencoder is closely related to the similarity of the data on which the autoencoder was trained. This limitation also reduces the scalability of these algorithms [51].
- ***Lossy output*** – In simplicity, this means that the compression (encode part of a autoencoder) and decompression (decode part of a autoencoder) operations reduce the performance of the neural network, leading to a less accurate representation compared to the input data [51].
- ***Problem with text and sequences*** – Autoencoders are designed to work with fixed length inputs [15].
- ***Sufficient amount of training data*** – The need to pay attention to a sufficient amount of training data depending on the complexity of the problem and the learning algorithm [14]. To find a sufficient amount of training data is an iterative problem that needs to be solved by empirical investigation (e.g. analysis of similar studies, use of domain expertise or statistical heuristics) [14]. Nonlinear algorithms usually capture more complex nonlinear relationships between input and output features [14].

These algorithms are often very flexible, more powerful, and we can present them with more random domain structures, but this performance is redeemed by more data than are required by linear algorithms [14]. In general, tens or hundreds of thousands training records for ”average” modeling problems and millions or tens-of-millions for ”hard” problems (e.g. complex deep learning) [14].

- ***An issue with the elementary autoencoder*** – Vanilla autoencoder model (a neural network with one hidden layer) is unable to randomly select the values for the decoder to generate the best possible data reconstruction (without explicit regularization, some points of the latent space are ”meaningless” once decoded) [52].

There is one specific type of autoencoder the Variational Autoencoder (VAE) that solves latent space organization [51][52]. VAEs are regularized versions of autoencoders making the generative process possible [51]. VAEs are better at catching properties of stochastic data and are better for generating new samples [52]. Still, the produced samples are often poor quality, mostly on the boundaries between two classes (for example, the shape between a circle and a triangle) [52]. VAEs are also limited with the fact, that they can only learn the properties of multiple Gaussian distributions, but what about, for instance, a uniform distribution [34]. The issue with the distribution described above can be solved by Adversarial Autoencoders (AAE) [34], but there are out of scope this work. The idea behind it is that you train the encoder to produce a latent space that looks like a prior distribution of your choice [34]. VAEs and AAEs types of autoencoders can be used to extend this work.

From this point, the section discusses the use of the autoencoder to measure data quality.

Based on the theoretical framework (see Section 1 about Theoretical framework), data quality measurement using an autoencoder can be used in the most straightforward application for structured data (see Section 1.1.2.2 about Types of data) at the aggregation level of the attribute (i.e. column values are input for the autoencoder) and record (i.e. summary of results from individual ”column” autoencoder within one record) (see Section 1.1.2.4 about General data characteristics).

Autoencoder should contribute to data quality measurement with greater flexibility, independence of domain knowledge and the ability to detect data quality deficiencies with greater automation. Therefore, this work will use the ability of the autoencoder to detect outliers in measuring data quality.

In current data quality tools, only simple outlier detection methods (e.g. box plots) are used compared to the current state of research [19]. The tools do not widely or at all support multivariate outlier detection or one of the more sophisticated approaches, such as z-scores, linear regression models or probability models [19]. Several tools are using Clustering (e.g. k-means) as an outlier detection method [19].

After training the autoencoder with the input data, the newly arriving data can be evaluated for a reconstruction error. The reconstruction error should be high enough for outlier data.

An autoencoder could be more independent on the type of data attribute when the input values will be converted into a text representation and then into a numerical representation, with the need to think about the semantic representation of the text to preserve the meaning of the converted symbols. One option is to use a Semantic data types detection (see Section 2.1 about Analysis of state-of-the-art approaches to enhance DQ) or NLP (Natural language processing) approaches, which are outside the scope of this work. Therefore, only fundamental approaches of a conversion text representation into a numerical representation will be applied in the experiment. More advanced approaches can be used to extend this work.

This solution can be partially ineffective on sparse data because the autoencoder learns at little informative values (e.g. null, NaN). Data with high heterogeneity is also not an adequate input. In this case, we may notice a high number of outliers, making this solution ineffective.

The approach described above requires to know the longest possible length of the encoded sequence of input data in advance.

This approach do not solve the relationships between other information in the table or database. The method only solves the adequacy of a single column value. We can determine how adequate the whole row is from the individual

results of the autoencoder for column values. The advantage of this approach is that it can locate the quality error within the column. In order to determine when a given whole row does not meet the quality, it is necessary to establish a metric that would numerically determine the quality by combining results for all columns or subset or columns of the given row. Such a metric could be set as the sum of all column reconstruction errors (e.g. MSE) per row. A detailed description with a figure (see Figure 3.1) of this approach can be found in the experiment section of this thesis.

This method for measuring data quality is expected to be able to detect fundamental shortcomings in data quality with an emphasis on reducing the need for domain knowledge (e.g. the correct format or structure of attributes) or human intervention. In practice, the work will experiment with this method for measuring data quality in autoencoder experiment (see Section 3.3.1 about Experiment 1 – Autoencoder).

## 2.3 Data quality measurement using Association Rule Mining

This section introduce some of the essential concepts related to association rule mining and present Apriori algorithm in the context of data quality measurement.

Association rule learning/mining is a rule-based machine learning method for discovering underlying relations between features in the data that can be expressed in the form of an `IF-THEN` rule [64][42].

Association rule mining can be formally defined as follows [64][5]:

- $I = \{i_1, i_2, \ldots, i_m\}$ is a set of `items` (e.g. milk, cherries, beer).
- $D = \{t_1, t_2, \ldots, t_n\}$ is a set of transactions, called a `transaction database`, where each transaction $t$ has `transaction ID` and contains a subset of the `items` in $I$.
- $X \Rightarrow Y$, where `itemsets` $X, Y \subseteq I$ and $X$, $Y$ do not intersect, is a `association rule` (more specific version [5] is $X \Rightarrow i_j$ for $i_j \in I$).

There are various metrics (e.g. Support, Confidence, Lift, Conviction) to achieve a selection of underlying rules from a set of all possible rules.

***Support*** refers to the frequency of an `itemset` $X$ in a `transaction database` $D$, and is defined as the proportion of transaction $t$ in the `transaction database` $D$ which contains the `itemset` $X$:

$$\texttt{supp}(X) = \frac{|X(t)|}{|D|}, (\text{range: } [0, 1]), \tag{2.1}$$

where $X(t) = \{t \text{ in } D | t \text{ contains } X\}$ [64][5]. In the example, if an `itemset` $X$ occurs $7$ times in a `transaction database` $D$ of $10$ transactions, then a support is $0.7$ since it occurs in $70\%$ in all transactions in the `transaction database` $D$.

An `itemset` $X$ in a `transaction database` $D$ that its support is greater than or equal to the minimal support threshold (`minsupp`) given by users or experts, is called a `frequent itemset` [64].

The support of the `association rule` $X \Rightarrow Y$ is the support of $X \cup Y$ [64].

***Confidence*** of an `association rule` $X \Rightarrow Y$ in the context of `transaction database` $D$ refers to the proportion of the transactions that contains $X$ which also contains $Y$ [64][42]. Specifically, confident is defined as [64]:

$$\texttt{conf}(X \Rightarrow Y) = \frac{\texttt{supp}(X \cup Y)}{\texttt{supp}(X)} = \frac{|(X \cup Y)(t)|}{|X(t)|}, (\text{range: } [0, 1]). \tag{2.2}$$

In the example, if a rule $X \Rightarrow Y$ has a support of $0.27$ since the rule occurs in $27\%$ of all the transactions. And if the rule $X \Rightarrow Y$ confidence is $0.74$, it means that for $74\%$ of all the transactions containing $X$ the transactions also contains $Y$.

Support (frequencies of occurring patterns [64]) and confidence (strength of implication [64]) are elementary quality measurements of the each `association rule` and measure how underlying the `association rule` is [5].

Just as we can set a minimal support threshold (minsupp) for a given association rule $X \Rightarrow Y$, thus we can condition the association rule with a minimal threshold confidence (minconf), which is given by users or experts, as well. An association rule $X \Rightarrow Y$ meeting the requirements of the described minimum thresholds is called a strong/valid rule and is defined as [64]:

1. $\mathtt{supp}(X \cup Y) \geq \mathtt{minsupp}$,
2. $\mathtt{conf}(X \Rightarrow Y) = \frac{\mathtt{supp}(X \cup Y)}{\mathtt{supp}(X)} \geq \mathtt{minconf}$.

***Lift*** is a quality measurements similar to the confidence but it also accounts for how popular $Y$ is [27]. The lift of the association rule $X \Rightarrow Y$ gives the correlation between $X$ and $Y$, and is defined as [27]:

$$\mathtt{lift}(X \Rightarrow Y) = \frac{\mathtt{conf}(X \Rightarrow Y)}{\mathtt{supp}(Y)} = \frac{\mathtt{supp}(X \cup Y)}{\mathtt{supp}(X) \times \mathtt{supp}(Y)}, (\text{range: } [0, \infty]). \tag{2.3}$$

The consequences arising from the lift value are [42]:

- If $\mathtt{lift}(X \Rightarrow Y) = 1$, then $X$ and $Y$ are independent and the association rule $X \Rightarrow Y$ can not be derived from the transaction database $D$.
- If $\mathtt{lift}(X \Rightarrow Y) > 1$, then $X$ and $Y$ are dependent and the association rule $X \Rightarrow Y$ can be considered as potentially useful, based on the degree of dependence.
- If $\mathtt{lift}(X \Rightarrow Y) < 1$, then the presence of $X$ in the association rule $X \Rightarrow Y$ have negative effect on $Y$.

***Conviction*** was developed as an alternative to confidence and is similar to lift [26]. In contrast to lift, it uses the information of the absence of $Y$ in `association rule` $X \Rightarrow Y$. Conviction of an `association rule` $X \Rightarrow Y$ in the context of `transaction database` $D$ refers to the probability that $X$ appears without $Y$ if they were dependent with the actual frequency of the appearance of $X$ without $Y$ [13][26]. In other words, conviction measures the expected error of an `association rule` $X \Rightarrow Y$ (i.e. How often $X$ occurs in `transaction database` $D$ where $Y$ does not.) [62].

$$\texttt{conv}(X \Rightarrow Y) = \frac{1 - \texttt{supp}(Y)}{1 - \texttt{conf}(X \Rightarrow Y)} = \frac{P(X) \times P(Y)}{P(X) \cup P(Y)}, (\text{range: } [0, \infty]). \quad (2.4)$$

The conviction values equal to $1$ indicates independence of the rule and rules that always hold have the the conviction values $\infty$ [26]. A high value therefore means that $Y$ depends strongly on $X$. The conviction values in $(0,\ 1)$ means that $X$ and $Y$ are independent. In example, the conviction value of $1.66$ tell us that the `association rule` $X \Rightarrow Y$ would be incorrect $66\%$ ($1.66$ times) more often if $X$ and $Y$ of the rule were independent [26].

In addition to the commonly used interest measures mentioned above, exists others [26] (see the link).

There are various association rule mining algorithms (e.g. AIS, SETM, Apriori, Aprioritid, Apriorihybrid, FP-growt) [39], for this work was chosen the most commonly used, ***Apriori algorithm*** [3] (see Algorithm 1). Apriori algorithm

uses BFS algorithm to find the support for of itemsets and generate candidates so that exploits the downward closure property of support [3].

**Algorithm 1**: Apriori algorithm [3].

**input** : Transaction database $D$, Support threshold minsupp
**output**: Frequent Itemsets

// $L_k$: Frequent itemset of size k
$L_1 \leftarrow \{\text{large 1-itemsets}\}$;
k $\leftarrow 2$;
**while** $L_{k-1} \neq \emptyset$ **do**
    // New candidates, $C_k$: Candidate itemset of size k
    $C_k \leftarrow \{c = a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a, \{s \subseteq c \mid |s| = k-1\} \subseteq L_{k-1}\}$;
    **for** transactions $t \in D$ **do**
        // $M_t$: Candidates in transaction $t$
        $M_t \leftarrow \{c \in C_k \mid c \subseteq t\}$;
        **for** candidates $c \in M_t$ **do**
            count[$c$] $\leftarrow$ count[$c$] $+1$;
        **end**
        $L_k \leftarrow \{c \in C_k \mid \text{count}[c] \geq \text{minsupp}\}$;
        k $\leftarrow$ k $+1$;
    **end**
**end**
**return** $\bigcup_k L_k$

Association Rule Mining typically consists of two parts [64][3]:

- ***Frequent itemsets generation*** – generate all frequent itemsets in a given transaction database $D$ based on support threshold minsupp. This is the Apriori algorithm (see Algorithm 1).
- ***Rule Generation*** – generate all strong association rules from the frequent itemsets that have support and confidence greater than the user-specified minsupp and minconf thresholds (or alternatively use other measures like Lift [42] and Conviction [26]).

***Limitation of Apriori algorithm*** is that it may suffer from large computational overheads when the number of frequent itemsets is very large because

it requires a full `transaction database` $D$ scan many times (i.e. finding candidate itemsets) [64][43]. Due to this, Apriori algorithm will be very low and inefficiency on the a large `transaction database` $D$ when memory capacity is limited, because the algorithm assumes that the database is permanent in the memory [43]. There are approaches and researches addressing these shortcomings of the Apriori algorithm (e.g. advanced pruning) [64][43]. Due to the scope of this work, these additional approaches will not be considered, but are noted, for the possibility of extending this work or experiments.

This method has been chosen for measuring data quality because of its ability to identify a set of underlying rules that collectively represent knowledge within data. The second reason is that data quality tools rarely support association rule mining because customers and vendors do not consider it as part of data quality [19]. No single tool analysed in the survey [19] offers an association rule mining approach.

In the following paragraphs, the Association Rule Mining will be placed in the context of data quality measurement. The Association Rule Mining can be used to find relationships (rules) between columns (multi-column profiling) [19], that can be seen in the theory from the beginning of the section.

The association rule extraction approach could find objective relationships based on the metrics presented above with significant independence from data knowledge or application context to developing useful data quality measurement. In an example, extracted strong association quality rules could proactively check data using information about their strength (e.g. using some of the metrics described above).

In this regard, we encounter the limitation that the Association rule mining approach is originally designed to work with qualitative and discrete data attributes (see Section 1.1.2.3 about Data attribute types) [46]. However, it is possible to discretize any continous data attributes into discrete intervals [46]. It is necessary to split the range of values into a suitable number of intervals during the discretization process [46]. The discretization process is critical in order to reduce the large amount of intervals to obtain high confidence rules [46]. Nevertheless, the too small intervals can lead to low support rules [46].

Several discretization approaches are available such as equal-width and equal-depth methods [46]. There are univariate or multivariate approaches to the discretization process [46]. As the name implies, the univariate discretization considers one continuous attribute at a time and is often used and simple while multivariate discretization considers simultaneously multiple attributes [46]. The multivariate discretization presents more advantages, especially in the case of discovering association rules, that look for a relationship across multiple attributes [46]. There are also science efforts to improve association rule algorithms, in terms of performance or reducing the set of rules with a focus on strong rules [40]. Appropriately rescaled input data (values of `Transaction database` $D$) should reduce interest in one of the numerical attributes by transforming all the numerical attributes on the same scale (e.g. Decimal Scaling, Min-Max Normalization, Z-score Normalization) [59].

The above findings show that the main challenge of this approach is data preprocessing. This greatly complicates the automation of this approach. Methods for data preprocessing should be chosen so that the way of automation is supported as much as possible.

In practice, the work will experiment with this approach for measuring data quality in Association rule mining experiment (see Section 3.3.2 about Experiment 2 – Association Rule Mining).

Chapter 3

# Experiments and Results

This chapter focuses on the practical application of the knowledge gained in the previous chapters for experiments aimed at measuring data quality with an emphasis on AI/ML-based techniques.

Data quality measurement experiments are in the field of Autoencoders (see Section 3.3.1 about Experiment 1 – Autoencoder) and Association Rule Mining (see Section 3.3.2 about Experiment 2 – Association Rule Mining).

The Section 3.1 describes the datasets used in the experiments. In the Section 3.2 is information about the technical environment on which the experiments were developed.

## 3.1 Datasets

For the following experiments, two datasets covering two classes of datasets were selected – consistent and inconsistent. The first dataset (see Section 3.1.1 about Consistent dataset - Macy's (e-commerce)) represents a class of datasets, where the values in a column are homogeneous (e.g. similar length of individual records, fewer missing records) and have a more transparent format. In contrast, the second dataset (see Section 3.1.2 about Inconsistent dataset - Open Food Facts) represents a class of inconsistent datasets whose values are mostly of different nature within a column (e.g. unclear formats, lengths of column values). To the best of my knowledge, there is no labelled dataset for data quality or data quality measurement. For this reason, both datasets were

enriched with synthetic data quality issues within the domain context of the dataset and data quality labels for experimental purposes.

**Table 3.1:** Datasets basic information.

| Dataset | # instances | # columns | Pandas data types[a] |
|---|---|---|---|
| Macy's | 40897 | 14 | object(#12), float64(#2) |
| Open Food Facts | 1356289 | 181 | object(#57), float64(#122), int64(#2) |

[a] `https://pbpython.com/pandas_dtypes.html`

### 3.1.1 Consistent dataset - Macy's (e-commerce)

Macy's dataset was found on the Kaggle datasets repository [31]. Macy's is an American department store[32] selling products such as clothing, footwear or accessories. The dataset contains innerwear and swimwear products. The dataset has 40897 records and 14 columns (e.g. `product_name`, `price`, `pdp_url`, `brand_name`, `rating`) in CSV format. More detailed information about the dataset can be found in Table 3.1 and in the Dataset Kaggle link.

### 3.1.2 Inconsistent dataset - Open Food Facts

Open Food Facts[33] is a free, open, collaborative project that gathers information and data on food products from around the world. The dataset contains information such as ingredients, allergens, nutrition facts and information that can be found on product labels. The dataset has 1356289 records and 181 columns (e.g. `code`, `created_datetime`, `product_name`, `countries`, `additives`, `glucose_100g`, `sodium_100g`, `energy_100g`) in TSV format. The dataset has a significant number of missing or inconsistent values, that is caused by the collaboration of users on the dataset. More detailed information about the dataset can be found in Table 3.1 and in the Open Food Facts data link. Information on the different fields for the CSV exports is available in the following link.

## 3.2 Experiment environment

The experiments were carried out on a server with 2x Intel Xeon E5-2650v4 CPU, 256GB DDR4 RAM, 2x Nvidia Tesla V100 16GB GPU, and Ubuntu 16.04

[31] https://www.kaggle.com/datasets
[32] https://www.macys.com/
[33] https://world.openfoodfacts.org/

operating system. All experiments were implemented in the Python[34] programming language using libraries that provide functionalities useful for machine learning experiments. Essential libraries for experiments include:

- Keras[35] – a deep learning API written in Python, running on top of the machine learning platform TensorFlow[36].
- NumPy[37] – a library for scientific computing with Python.
- Pandas[38] – a library for data analysis and data manipulation with Python.
- Matplotlib[39] – a library for creating static, animated, and interactive visualizations in Python.
- Natural Language Toolkit (NLTK)[40] – a set of libraries for symbolic and statistical natural language processing (NLP).

For prototyping the experiments JupyterLab[41] was used. JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data.

Conda[42] package manager and environment management system is used for easier reproducibility of the experiments. All libraries and their versions are available in Conda environment (see Appendix files list – `environment.yml`) .

## 3.3 Data quality measurement experiments

This section follows up on the previous theoretical chapter (see Section 2 about Data quality measurement methods) and comes with the implementation of novel approaches in the field of data quality measurement – Autoencoders (see Section 3.3.1 about Experiment 1 – Autoencoder) and Association Rule Mining (see Section 3.3.2 about Experiment 2 – Association Rule Mining).

[34] https://www.python.org
[35] https://keras.io/
[36] https://www.tensorflow.org/
[37] https://numpy.org/
[38] https://pandas.pydata.org/
[39] https://matplotlib.org/
[40] https://www.nltk.org/
[41] https://jupyter.org/
[42] https://docs.conda.io/

### 3.3.1 Experiment 1 – Autoencoder

The following experiment follows up on the section (see Section 2.2 about Data quality measurement using Autoencoder) where the theoretical basis of the autoencoder was introduced as an approach for measuring data quality.

This experiment analyzes the potential of an autoencoder to measure data quality. The experiment will be measured on the two datasets mentioned above (see Section 3.1 about Datasets). Before describing the application and summarizing the results of this approach for measuring data quality, the experiment autoencoder model for this approach and its hyperparameters will be presented. An analysis and comparison with the complementary non-AI method will describe at the end of this experiment.

An elementary autoencoder model (see Code 3.1) was chosen for essential prototyping in order to detect anomalies. The output they are trying to generate is a reconstruction of the received input. During this reconstruction, a reconstruction error may occur, which is measured by normalized MSE.

```
1  def create_ae_model(input_dim, enc_dim, latent_dim):
2
3      input_layer = Input(shape=(input_dim, ))
4      encoder = Dense(enc_dim, activation="tanh",
5                      activity_regularizer=regularizers.l1(10e-5))(input_layer)
6      encoder = Dense(latent_dim, activation="relu")(encoder)
7      decoder = Dense(enc_dim, activation='relu')(encoder)
8      decoder = Dense(input_dim, activation='tanh')(decoder)
9
10     return Model(inputs=input_layer, outputs=decoder)
```

**Source code 3.1:** Experiment 1 – Autoecoder model.

An autoencoder model is trained for each dataset feature. This setting allows us to record the reconstruction error for the desired value in a feature as well as the total reconstruction error for an entire row (sum of reconstruction errors for the entire record) (see Figure 3.1). The reconstruction error is normalized within the features. As part of the extension of the work, it would be appropriate to deal with additional metrics for the entire record. If we give a well-trained

autoencoder an inappropriate value (e.g. wrong format), we will probably get a significant reconstruction error.

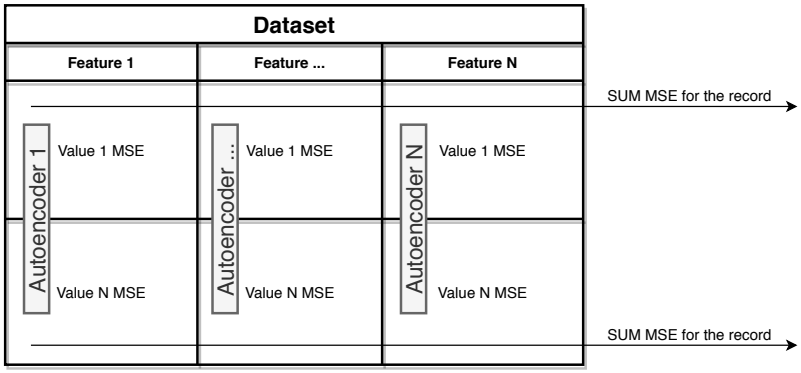


**Figure 3.1:** Experiment 1 – Experiment design with autoencoders.

Because the data attributes are heterogeneous (text and numbers), all dataset attributes have been converted to a string representation. The preparation of the text for the autoencoder was performed by tokenizing the value of the feature in order to preserve the elementary semantic information of the text. Subsequently, each token was converted to a numerical representation using UTF16. These numerical representations of tokens were composed into the resulting autoencoder model input value representation so that zero value was inserted between every two numerical tokens. In an example, value `'Hi Michael'` is represented as `[72, 105, 0, 77, 105, 99, 104, 97, 101, 108]`. Before submitting the modified input to the autoencoder, the data is scaled using the $MinMaxScaler$ method and divided into a training and test set. This part of the experiment (a conversion of text into numerical representation) is a potential candidate for the work extension.

There are several ways to design an autoencoder. Several iterations of the autoencoder design were performed (e.g., using the $sigmoid$ activation function instead of $tanh$) to determine the most appropriate design and hyperparameters (epoch count – 30, batch size - 256) for the experiment (see Codes 3.1).

The basic parameter for measuring data quality is MSE threshold, which in this case is set at the quantile $0.9999$. Values that have the reconstruction error value greater than the specified threshold are considered as outliers.

#### 3.3.1.1 Autoencoder experiment and results – Macy's dataset

Synthetic errors in the domain context were created as part of an experiment on the Macy's dataset. For instance, these are the following types of synthetic errors – product description in the wrong language, wrong product price format, inadequate product category and wrong product URL domain. More specifically, synthetic errors will be assessed at the end of this section.

As already described in the design of the experiment (see Figure 3.1) an autoencoder model was trained for each dataset feature. The performance of the autoencoder models seems to perform well for each feature because they are able to minimize the loss function within epochs meaningfully. All performance comparisons of the models are attached in the appendix (see Appendix files list – `Experiment-1-autoencoder-macys-loss-columns.pdf`) . One representative of the model performance graph was chosen for the example, specifically for the feature '$product\_name$' (see Figure 3.2).

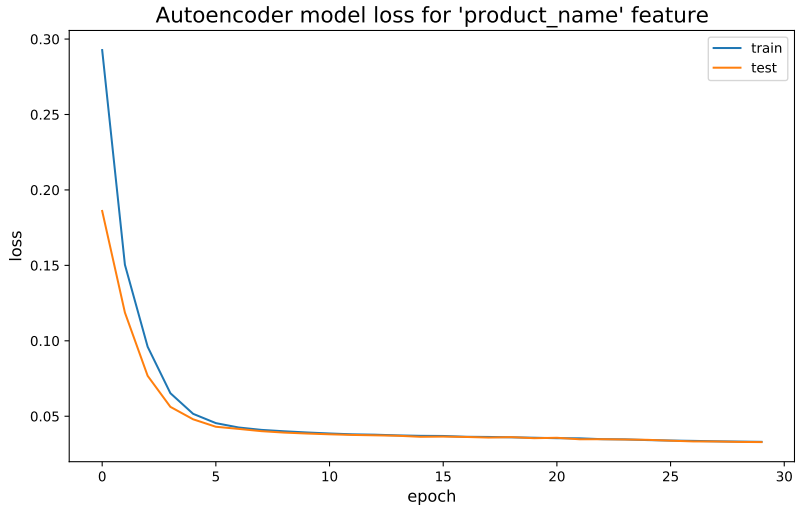


**Figure 3.2:** Experiment 1 – Macy's dataset – Autoencoder model performance for Product name feature.

In the histogram (see Figure 3.3), we can see the distribution of the sum of MSE for the record in percentage and its MSE threshold (quantile 0.9999). Nine records were evaluated as outliers. These are records that do not qualitatively correspond to the dataset standard. Within these five outliers, two synthetically created data quality errors were found. The remaining seven nonsynthetic records are suspected of insufficient data quality and are suitable candidates

for the investigation of domain experts. Thus, both records with artificially created defects in data quality and inadequate records that were part of the original dataset were found. For the comparison, in the Figure 3.4) we can see columnar MSE for synthetic and nonsynthetic record. In the synthetic record (Record id: 19386, see title in the Figure 3.4), the MSE for the synthetically generated data quality error greatly predominated.

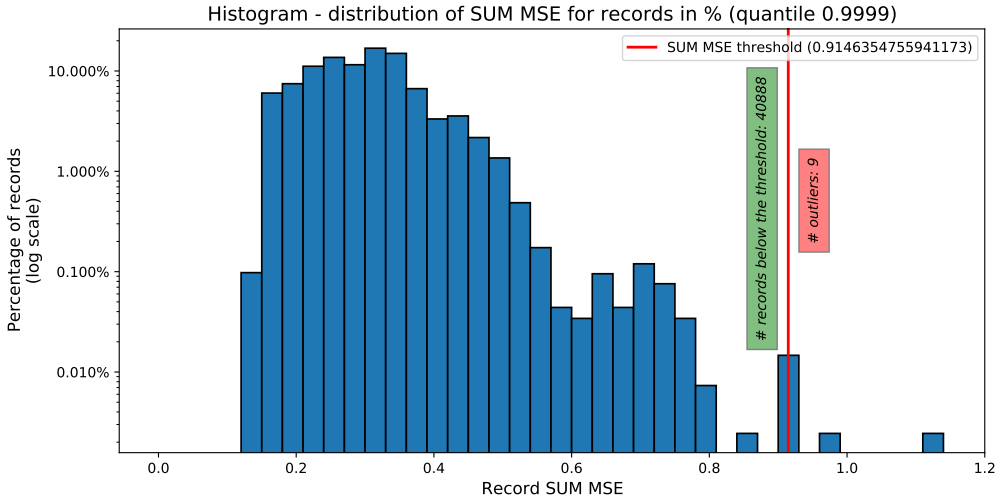


**Figure 3.3:** Experiment 1 – Macy's dataset – Histogram – Distribution of SUM MSE for records in % (quantile 0.9999).

The seven non-synthetic records are almost identical, except for the feature `available_size` in the case of the one record (Record id 27814)(see Figure 3.4), that differs in a one list value ($34C$ instead of $32G$ size). This value was evaluated as an outlier in the record. These seven records appear to be qualitatively suspicious due to their duplication and value such as:

- `style_attributes` feature – `"Please select specific item for product information.", "Imported", "Web ID: 1472235"` (For example, a more common value is: `"Wireless cups", "Wide band below bust with logo", "Racerback with sheer mesh panel"`)

It should be mentioned that the columns `total_sizes` and `review_count` also have a large impact in the selection of these records among SUM MSE outliers. At least a values 305 for `review_count` features appear to have an inadequate column MSE value. The value does not seem to be bad to determine the number

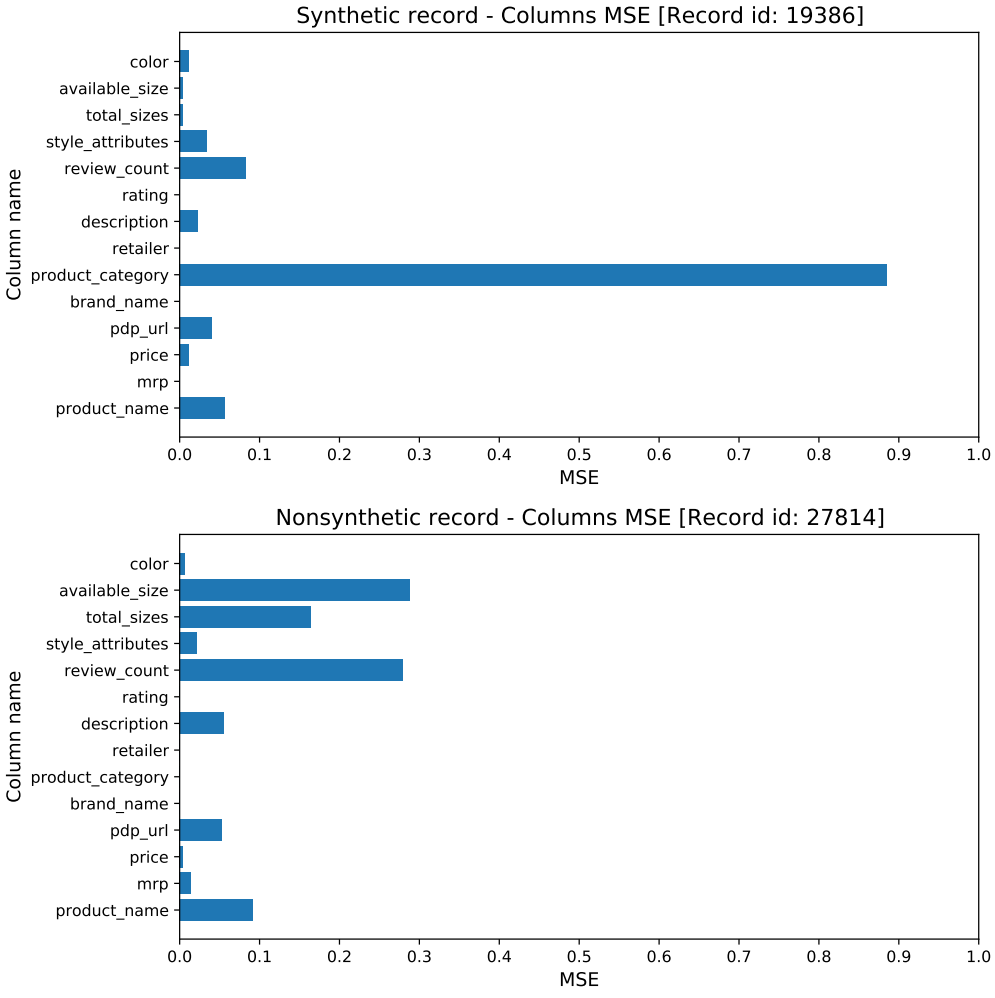


**Figure 3.4:** Experiment 1 – Macy's dataset – Synthetic vs. non-synthetic record – Columns MSE.

of reviews of the goods. The reason for the large MSE for this value may be that the value $305$ appears in the dataset only $10$ times.

As mentioned in the experiment theory section (see Section 3.3.1 about Experiment 1 – Autoencoder), as an extension of this experiment, it would be useful to explore additional metrics that would assess whether a given entire record is a suitable adept for data quality control. For example, information about the frequency of a given value in a dataset could be included in the metric.

Autoencoder model found records that are adequate adepts that should be examined by data quality experts.

Histograms of the MSE distribution and threshold for individual features of the dataset are available in the appendix (see Appendix files list – `Experiment-1-autoencoder-macys-mse-dist-columns.pdf`) .

Of the total number of the records (per attribute) were approximately $0.00209\%$ of non-synthetic data quality errors evaluated as outliers.

The success rate of autoencoder models on synthetic data quality errors was $56.25\%$ (see Table 3.2) (MSE Threshold (quantile 0.9999)).

**Table 3.2:** Experiment 1 – Success of autoencoder models on synthetic data quality errors – Macy's dataset.

| **Category name** | $\sum$ |
|---|---|
| # synthetic DQ issues | 16 |
| [synthetic] # feature records below MSE threshold | 7 |
| [synthetic] # feature records above MSE threshold | 9 |

The following list contains the essential outliers found in this experiment (# records: $40897$):

- Feature $price$ (# unique values: $177$)(correct example format: $\$20.66$) – Detection of the wrong product price format (£$16.50$, $\$16$).
- Feature $pdp\_url$ (# unique values: $1242$) – Detection of the wrong domain in the URL ($.uk$ instead of $.com$).

- Feature $brand\_name$ (# unique values: $9$) – Detection of a domain mismatched value for a product brand ('Coca-Cola').
- Feature $product\_category$ (# unique values: $7$) – Detection of a domain mismatched value for a product category ('Headphones').
- Feature $retailer$ (# unique values: $2$) – Detection of a domain mismatched value for a product category ('Kaufland US').
- Feature $description$ (# unique values: $591$) – Detect the use of inappropriate language for product description (German instead of English).
- Feature $rating$ (# unique values: $28$) – Detection of synthetic data quality errors for a rating that was out of range (e.g. $9.0$ and $6.0$) (correct range: $0.0$–$5.0$).
- Features $style\_attributes$ (# unique values: $854$), $available\_size$ (# unique values: $612$), $color$ (# unique values: $491$) – For each of these features, from two to five values were evaluated as outliers. Domain knowledge is required to determine that these values are erroneous.

The following list contains the significant synthetically created errors that were not found by this experiment:

- Feature $price$ – Failure to detect synthetic data quality errors for product price such as $\$24,50$ (wrong format), $\$0.50$ (low price) and suspicious value ($\$0.0$).
- Feature $description$ – Failure to detect synthetic data quality errors for product description that was too short and concise.
- Feature $color$ (# unique values: $491$) – Failure to detect synthetic data quality error for product color ('blakc').
- Feature $total\_size$ (# unique values: $464$) – Failure to detect synthetic wrong format ({ } instead of [ ])(correct example format: `["S", "M", "L"]`).

A summary of the number of records above and below the MSE threshold (synthetic, non-synthetic) and the number of unique values for each feature

can be seen in appendix (see Appendix files list – `Experiment-1-autoencoder-macys-mse-summary-results.csv`) .

#### 3.3.1.2 Autoencoder experiment and results – Open Food Facts dataset

The Open Food Facts dataset has been reduced to 339072 records due to its size. Domain synthetic errors of a similar format as the lead dataset were generated on the reduced dataset. Significant synthetic errors will be listed at the end of this section.

The performance of the autoencoder models seems to be worse than in the previous case. The performance of the models could be divided into three categories (see Figures 3.5, 3.6 and 3.7).

Some autoencoder models work well (see Figure 3.5) because they are able to meaningfully minimize the loss in epochs. The poor performance of some models is probably caused by a small/large number of unique values or NaN and inconsistencies in values. All performance comparisons of the models are attached in the appendix (see Appendix files list – `Experiment-1-autoencoder-foods-loss-columns.png`) .

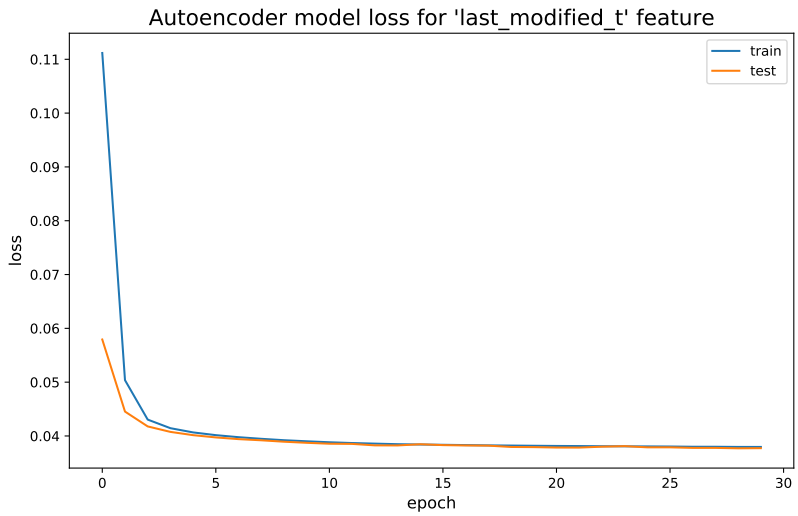


**Figure 3.5:** Experiment 1 – Open Food Facts dataset – Autoencoder model performance for Code feature.

Inadequate performance of the autoencoder model for the *packaging* feature (see Figure 3.6) can be caused by lots of NaN and unique values (total number of values: 339072, # NaN: 285686, # unique values: 12989).

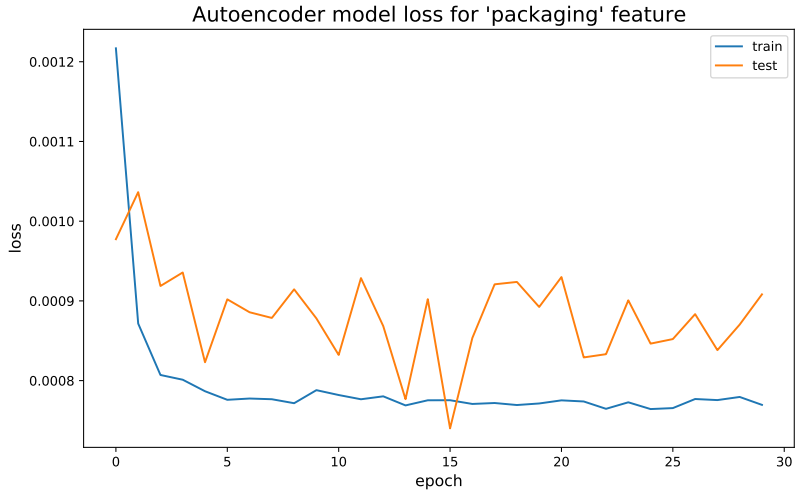


**Figure 3.6:** Experiment 1 – Open Food Facts dataset – Autoencoder model performance for Packaging feature.

Inadequate performance of the autoencoder model for the *allergens_en* feature (see Figure 3.7) is due to the fact that all values of this feature are NaN.

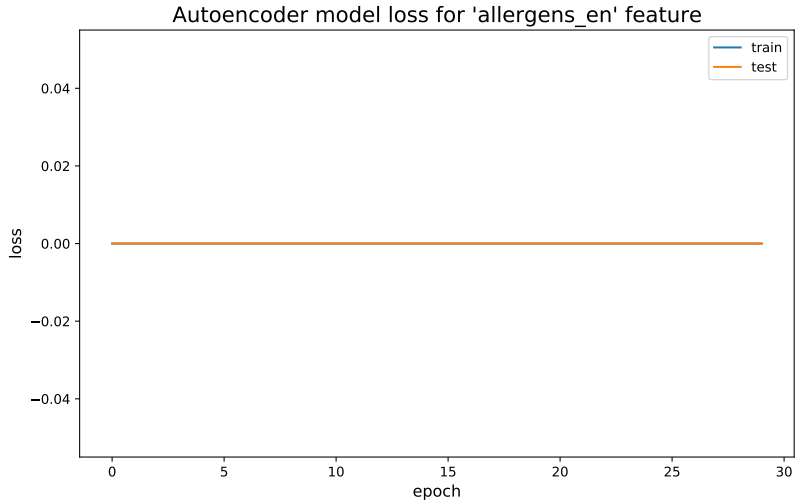


**Figure 3.7:** Experiment 1 – Open Food Facts dataset – Autoencoder model performance for Allergens en feature.

In the histogram (see Figure 3.8), we can see the distribution of the sum of MSE for the record in percentage and its MSE thresholds (quantile 0.9999). Thirty four records were evaluated as outliers. These are records that do not qualitatively correspond to the dataset standard. Within thirty four outliers, no synthetically created data quality errors were found.

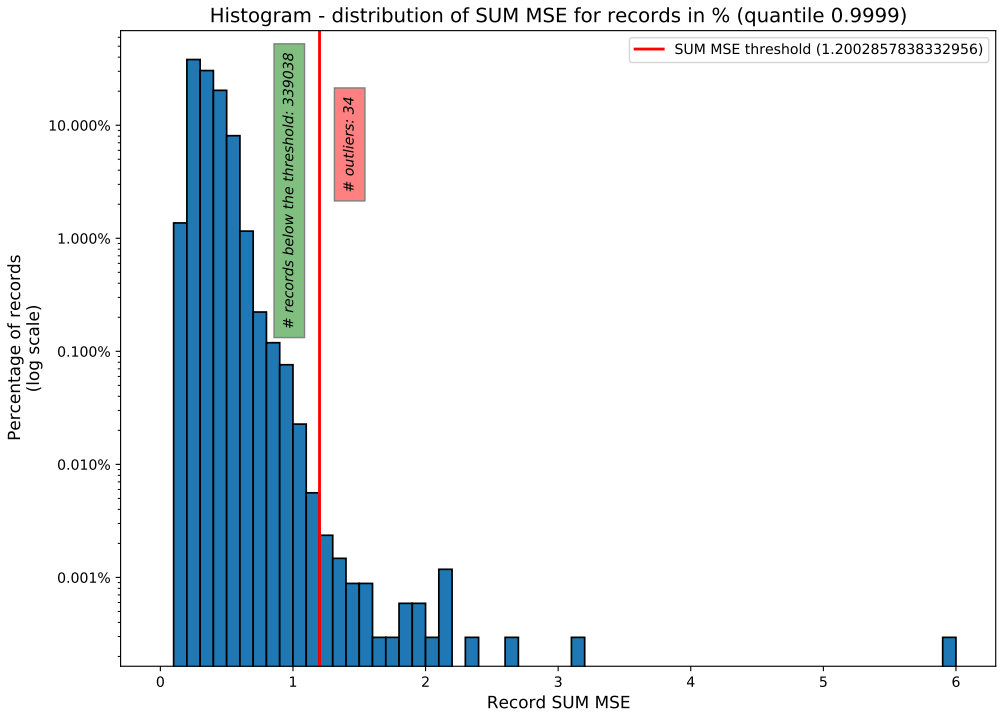


**Figure 3.8:** Experiment 1 – Open Food Facts dataset – Histogram – Distribution of SUM MSE for records in % (quantile 0.9999).

An interesting case is the record outlier (see Figure 3.8) with a large SUM MSE value (5.930). The outlier has anomalies in 6 URL columns (e.g. $image\_url$, $image\_small\_url$, $image\_ingredients\_url$). Because all the URLs have similar anomaly values, the one anomaly URL will be presented:

Anomaly case of the the record feature $image\_url$ is:

`https://static.openfoodfacts.org/images/products/019/542/500`
`/2492345310300039515190614102l8/front_fr.3.400.jpg`

Non-anomalous case of record feature $image\_url$ is:

`https://static.openfoodfacts.org/images/products/340/159/671/6356/front_fr.4.400.jpg`

For the anomalous URL, it can be seen that its penultimate part looks suspicious at first glance. This record should be analyzed by data quality domain experts.

Histograms of the MSE distribution and threshold for individual features of the dataset are available in the appendix (see Appendix files list – `Experiment-1-autoencoder-foods-mse-dist-columns.png`) .

Of the total number of the records (per attribute) were approximately $0.00666\%$ of non-synthetic data quality errors evaluated as outliers.

The success rate of autoencoder models on synthetic data quality errors was $43.75\%$ (see Table 3.3) (MSE Threshold (quantile 0.9999)) as in the previous case.

**Table 3.3:** Experiment 1 – Success of autoencoder models on synthetic data quality errors – Open Food Facts dataset.

| **Category name** | $\sum$ |
|---|---|
| # synthetic DQ issues | 16 |
| [synthetic] # feature records below MSE threshold | 9 |
| [synthetic] # feature records above MSE threshold | 7 |

The following list contains the essential synthetic outliers found in this experiment (# records: $339072$):

- Feature $created\_datetime$ (# unique values: $309116$) – Detection of bad data formats (’`2017-07-26T18:27:10`’, ’`2017/07/26 18:27:10`’, ’`2019-10-02T11:00:13.155`’). From non-synthetic dates were 31 dates were considered to be outliers, but they were in the correct format. These were mostly older dates (e.g. ’`2012-05-29T18:09:00Z`’) or dates around midnight (e.g. ’`2020-01-01T00:51:07Z`’).
- Feature $salt\_100g$ (# unique values: $6915$) – Detection of wrong data types (string instead of numeric data).
- Feature $nutrition\_score\_fr\_100g$ (# unique values: $55$) - Wrong format detection (added wrong unit to numeric value – ’$15.0$ l’).

- Feature $nutriscore\_grade$ (correct values: `{'a', 'b', 'c', 'd', 'e', 'nan'}`) (# unique values: 7) – Detection of inadequate score value (9).

- Feature $code$ (# unique values: 339027) – Detection of the specific code value ('841-010-000-2002').

The following list contains the significant synthetically created errors that were not found by this experiment:

- Feature $serving\_quantity$ (# unique values: 1229) – Failure to detect a negative value for quantity ($-86.6$). (The error was not detected due to selected tokenization filters in the experiment, where one filter character was a hyphen.)

- Feature $last\_modified\_datetime$ (# unique values: 293655) – Failure to detect wrong time format, when the value 72 was specified in the hour section ('`2019-10-15T72:21:10Z`').

- Feature $image\_url$ (# unique values: 235001) – Failure to detect the URL referring to a file with a `.txt` extension instead of a `.jpg` extension. (As an extension of this algorithm, Python parsing module of the library Urllib[43] could be used to more accurately represent the semantics of URL features.)

- Feature $countries$ (# unique values: 35062) – Failure to detect city name ('Prague') in the feature countries.

- Feature $main\_category$ (# unique values: 11307) – Failure to detect numeric value (16489.456) in text feature.

- Feature $fiber\_100g$ (# unique values: 1139) – Failure to detect too high a numerical value (1562.3) for the feature in the value range ($-6.7$, 439.0).

A summary of the number of records above and below the MSE threshold (synthetic, non-synthetic) and the number of unique values for each feature can be seen in appendix (see Appendix files list – `Experiment-1-autoencoder-foods-mse-summary-results.csv`) .

[43] `https://docs.python.org/3/library/urllib.html`

#### 3.3.1.3 Experiment and results for the complementary non-AI method

As a partial complementary non-AI method, checking the column format using a regular expression was chosen. To the best of my knowledge, there is no entirely complementary method to replace the data quality measurement method described above using the autoencoder. The experiment was performed on the product price feature from Macy's dataset.

The regular expression approach found all the data quality issues for the product price feature, but at the cost of data analysis, domain and technical knowledge, and manual effort to create a regular expression. The manual approach for a creating of the regular expression could be replaced by an automated approach (see Section 2.1 about Analysis of state-of-the-art approaches to enhance DQ).

### 3.3.2 Experiment 2 – Association Rule Mining

The following experiment follows up on the section (see Section 2.3 about Data quality measurement using Association Rule Mining) where the theoretical basis of Association Rule Mining was introduced.

The area of measuring data quality also includes the creation of business rules [19]. Creating these rules is often subject to a manual process. Therefore, this experiment will focus on the potential use of the Apriori algorithm to reduce manual effort to extract business rules. The automation of this approach will be supported by the NLTK library, which focuses on statistical processing of natural language (NLP) (see Section 3.2 about Experiment environment).

For this experiment, Macy's dataset(see Section 3.1.1 about Consistent dataset - Macy's (e-commerce)) was chosen as the dataset with business characteristics, as it is a dataset from the E-commerce domain. Within the domain, the experiment will attempt to answer the question: `"What are the relationships between the major products sizes?"`. For example, if the product has a size of "S" and "L", it also has a size of "M".

The experiment will process the text feature $total\_sizes$ to answer the above mention question. The feature can contain the following value, for example:

- `["32C", "32D"", "32DD", "S", "M"]`

All values of this feature were joined into one a text corpus in order to perform fundamental frequency analysis, to determine the major products sizes within the corpus/feature. This step partially replaces the manual approach.

An experiment dataset with indicator features (i.e. features containing $0$ or $1$ indicator value) was created based on the found major products sizes. For example, the feature name $total\_sizes\_contains\_L$.

Subsequently, each record was analyzed to check if it contains any of the found major product sizes. If so, a value of $1$ was set to the adequate feature (e.g. $total\_sizes\_contains\_L$) and record. Otherwise, a value of $0$ was set.

The experiment dataset was subsequently reduced to records containing at least two major product sizes values per record. In an example, if a record contains at least two major product sizes (e.g. "L", "M") in $total\_sizes$ feature, it has been included into the experimental dataset. This process reduced the number of dataset records for the next phase of the experiment.

The next phase of the experiment is the application of the implemented Apriori algorithm (see Algorithm 1) to find the Association Rules on the experiment dataset containing indicator features. The last phase of the experiments is an evaluation and a visualization of the results.

#### 3.3.2.1 Association Rule Mining experiment and results

In Figure 3.9 we can see a visual frequency analysis of the text corpus of products sizes (via WordCloud[44] library), which gave us a basic insight into the major values. Even such a basic visualization could give experts fundamental insights into the creation of business rules for a given domain. The Figure shows that we could consider the following values as the major products sizes – '34C', '34D', '36C' a '36D'.

The text corpus was subjected to a more detailed analysis using NLTK library for symbolic and statistical natural language processing (NLP). To preprocessing the text corpus into a well-defined sequences of linguistically-meaningful corpus units were used the following text preprocessing tasks:

- Tokenization – groups sequences of characters into logical elements called tokens (in the case of the experiment into words).

[44] `https://github.com/amueller/word_cloud`


**Figure 3.9:** Experiment 2 – Macy's dataset – WordCloud for the feature 'Total Sizes'.

- Filtration of punctuation, space characters and stop words (seldom provide any interesting information – e.g. 'a', 'the', "it").
- Lematization – finds lemma (canonical form) for a given token (e.g. 'were' -> 'be').
- Stemming – finds root form of a token (e.g. 'sitting' -> 'sit')

Frequency analysis of tokens was performed on such the preprocessed textual corpus with the following result in the Table 3.4 (the experiment was set up to search for 5 major products sizes). The more detailed analysis of the text corpus brought two new major products sizes – 'L' and 'XL'. The value of '34D' did not fit into the top five major products sizes.

**Table 3.4:** Experiment 2 – Token Frequency Analysis – Macy's dataset – Total sizes feature.

| Major products size | Frequency in the textual corpus |
|---|---|
| 'L' | 20426 |
| 'XL' | 13219 |
| '36C' | 10734 |
| '36D' | 10168 |
| '34C' | 9742 |

For the five top product sizes found above, the experiment dataset was created with the following indicator features:

- $total\_sizes\_contains\_L$,
- $total\_sizes\_contains\_XL$,

- $total\_sizes\_contains\_36c$,
- $total\_sizes\_contains\_36d$,
- $total\_sizes\_contains\_34c$.

The experiment dataset was subsequently reduced to records containing at least two major product values per record. This process reduced the number of dataset records from 40897 to 23964.

The influence of the minimum support on the number of frequented itemsets was analyzed in the following part of the experiment. Based on Figure 3.10, the expert should determine a minimum value for the support (Support threshold `minsupp`) in this experimental setup. This is a fundamental input parameter of this approach, but also of the Apriori algorithm itself (see Algorithm 1 inputs).

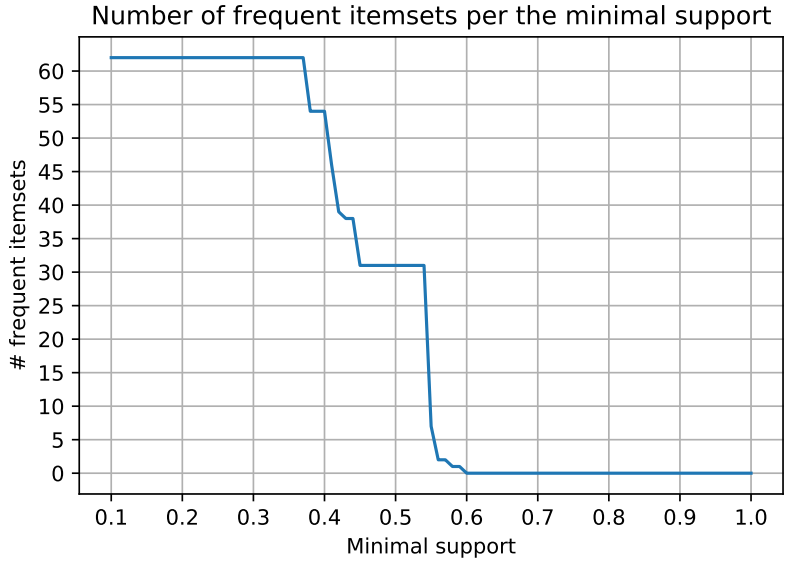


**Figure 3.10:** Experiment 2 – Macy's dataset – Number of rule for the minimal support.

In the last phase of the experiment, the Apriori algorithm was run with the minimum support (`minsupp`) value of 0.55. In Figure 3.11 we can see the found association rules and their metrics.

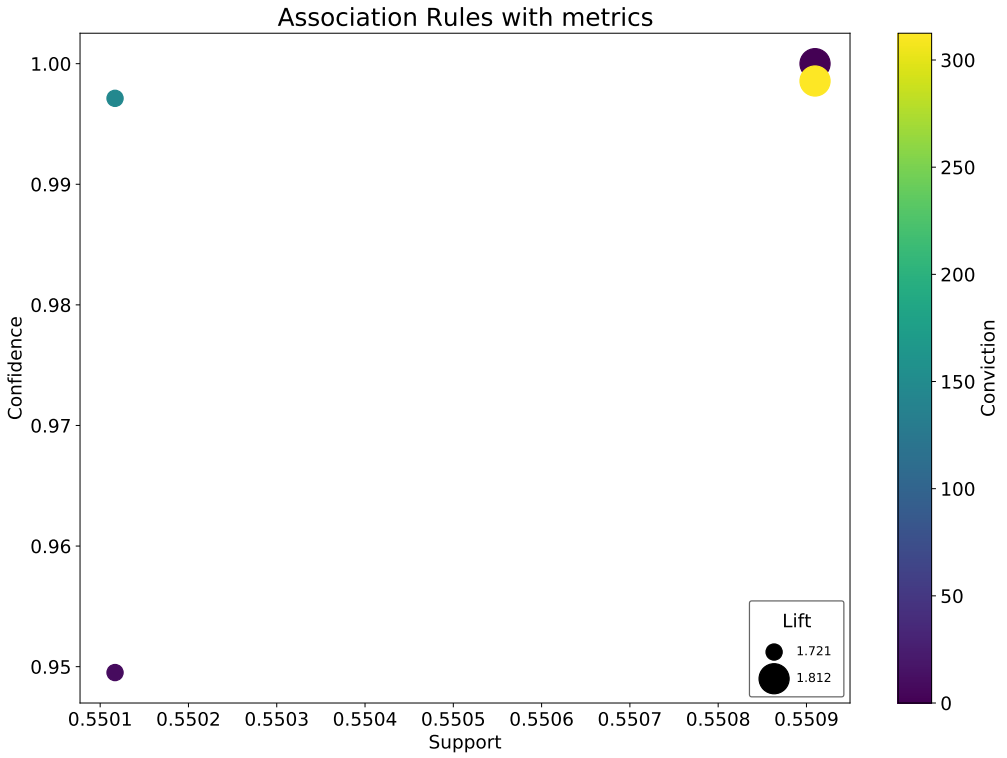


**Figure 3.11:** Experiment 2 – Macy's dataset – Final Association Rules with metrics.

The following summary is a detailed description of the found association rules and their consequences that are describing the relationships between the significant products sizes:

- Rule ($X \Rightarrow Y$):
  - `['total_sizes_contains_XL:1']` $\Rightarrow$ `total_sizes_contains_L:1`
  - Support: $0.55090$
  - Confidence: $1.0$
  - Lift: $1.8125$
  - Conviction: $0.0$

  - ***Consequence***: When the sizes of the product contain 'XL', it also contain the size of the product 'L'. The the product sizes containing 'XL' does not occur in transactions without the product size 'L'.

- Rule ($X \Rightarrow Y$):
  - `['total_sizes_contains_L:1']` $\Rightarrow$ `total_sizes_contains_XL:1`
  - Support: $0.55090$
  - Confidence: $0.9985$
  - Lift: $1.8125$
  - Conviction: $312.495$
  - ***Consequence***: When the sizes of the product contain 'L', it also often contains the size of the product 'XL', but also exists transaction with the product sizes containing 'L' without the product size 'XL'.

- Rule ($X \Rightarrow Y$):
  - `['total_sizes_contains_L:1']` $\Rightarrow$ `total_sizes_contains_36d:0`
  - Support: $0.55011$
  - Confidence: $0.9971$
  - Lift: $1.7210$
  - Conviction: $146.346$
  - ***Consequence***: When the sizes of the product contain 'L', it also often contains the size of the product '36D', but also exists transaction with the product sizes containing 'L' without the product size '36D'.

- Rule ($X \Rightarrow Y$):
  - `['total_sizes_contains_36d:0']` $\Rightarrow$ `total_sizes_contains_L:1`
  - Support: $0.55011$

- Confidence: 0.9495
- Lift: 1.7210
- Conviction: 8.878
- ***Consequence***: When the sizes of the product contain '36D', it also often contains the size of the product 'L', but also exists transaction with the product sizes containing '36D' without the product size 'L'.

This experiment revealed interesting relationships between significant product sizes, but considerable effort was required to preprocess the data to pass the appropriate input (`transaction database`) to the Apriori algorithm.

Domain experts can create rules for regular data quality monitoring based on the found association rules. The NLP approach used in the experiment greatly facilitated the process of preprocessing the data into the final form. The complexity of data preprocessing for Association Mining algorithms may be one of the reasons why this approach is not widely supported by data quality tools [19].

#### 3.3.2.2 Comparison with a complementary non-AI approach

The nearest complementary approach to the performed experiment is the manual definition of business rules with knowledge of domain requirements. Data quality tools usually implement the creation and application of business rules, where for some cases of data (e.g. zip codes), there are predefined rules (e.g. validation of addresses, data types) [19].

In tools such as Informatica Rule Builder, it is possible to define a set of conditions (IF-THEN statement) that must be met [33]. Alternatively, it is possible to select actions that will be taken if the conditions are not met [33]. Tools such as Informatica Rule Builder implement a graphical environment for defining business rules. It is a sophisticated tool with various functions for defining rules and requires considerable expertise and domain knowledge for business rules to be adequately implemented.

The experiment performed above is specialized for a specific type of textual data. Thus the experiment was able to implement partial automation at the expense of narrowing the problems.

Knowledge of the data semantics is essential information for the possibility of implementing automation in the field of defining business rules. The approaches described in Section 2.1 could be applied (Automatically Generating Regular Expressions via Genetic Programming, Semantic data types detection) to determine the semantic structure of the data or the data format.

Subsequently, rules could be generated based on the found semantic information (e.g. a regular expression for the data format) or the information about semantic data structure could be passed to an associated algorithm addressing the data quality. In an example, rules could be found for individual parts of a data format (e.g. the part of the data format intended for hours must be in the range of values between $0-23$). It could help the algorithm to simplify usage of automation process. These approaches could be integrated into the experiment as an extension of this work.

# Conclusion

In the first part of the thesis, the key aspects of data, data quality and data quality tools were analyzed. A fundamental observation of data is that the high degree of diversity and complexity brings abundant data types and complex data structures, thereby increasing the difficulty of automatization in the data quality filed.

To the best of my knowledge, a new categorization of data quality tools was identified within the analysis of data quality tools. This categorization divides data quality tools into two categories, according to the target group of their users. The category of general-purpose data quality tools is aimed at less technical users, which can bring unsolicited complexity into its implementation.

The second category focuses primarily on data engineering teams that are expected to have higher technical knowledge. This new category of tools can avoid complex solutions due to omitting user features. A tool of this type offering a wide range of ML-based methods to support automation may present a potential void in the data quality tools market. Current general-purpose data quality tools do not take full advantage of the potential of ML-based methods.

The second part of the thesis deals with ML-based state-of-the-art methods which have the potential to address data measurement in an innovative way and are not part of the implementation of data quality tools.

Semantic data type detection (e.g. date, URL, country) can have a significant impact on the automation of ML-based methods. For example, a text feature could be adequately tokenized according to the semantic information contained in the value.

Automatically generating Regular Expressions via Genetic Programming has the potential to measure data quality in the field of automatic measurement of a number of wrong format records.

One of the most promising approaches to detecting text duplicates may be the development of a machine learning model with NLP approaches (Word2Vec, Bag-of-Words, TF-IDF or Fuzzy string matching).

An approach with elementary Autoencoder models was chosen in order to detect anomalies. An autoencoder model is trained for each dataset feature. This setting allows us to record the reconstruction error for the desired value in a feature as well as the total reconstruction error for an entire row. The input values of the autoencoder were tokenized text values converted to a numerical representation. This approach was able to detect approximately half of the manually created synthetic data quality defects on both datasets. Several potential data quality defects contained in the original dataset were found as well. The advantage of this approach is in its automation (only one essential parameter for defining the reconstruction error threshold is needed) and in its application on heterogeneous attributes. The alternative non-AI approach, where a regular expression was defined for format checking, has found all wrong records in a given data feature, but is task-dependent, requires expert knowledge and manual effort. As part of the extension of the Autoencoder approach, it would be appropriate to implement more advanced Autoencoders models (e.g. VAE) and additional metrics that would assess whether a given entire record is a suitable adept for data quality control as well as with other methods for semantic representation which would adequately represent the structure autoencoder input features. To the best of author's knowledge, this approach is not described in any existing literature nor it is implemented in existing data quality tools.

In the Association Rule Mining approach was chosen Apriori algorithm because of its ability to identify a set of underlying rules that collectively represent knowledge within input data. The NLP approach used in the experiment greatly facilitated the process of data preprocessing. This approach was able to extract business rules for a given business question but required preprocessing the data into the required transaction dataset. The complexity of data preprocessing for Association Mining algorithms may be one of the reasons why data quality tools do not widely support this approach. This approach seems challenging to

automate. The similar alternative approach to the performed experiment is the manual definition of business rules which is entirely dependent on the skills of a domain expert.

Autoencoders and Association Rule Mining using NLP approaches were conducted as experiments on real-world, publicly available datasets in the practical part of the thesis. Both methods were compared with a similar alternative non-AI approach.

To conclude this thesis, a significant part of the thesis deals with data quality theory which provided a comprehensive insight into the field of data quality. The most significant contribution of this thesis is the finding of innovative approaches in the data quality measurement. Two of these state-of-the-art approaches were conducted as experiments.

# Bibliography


1. ACKOFF, Russell L. From data to wisdom. ***Journal of applied systems analysis***. 1989, vol. 16, no. 1, pp. 3–9.
2. ADLER, Mortimer J. ***A Guidebook to Learning: For a Lifelong Pursuit of Wisdom***. MacMillian, 1986. isbn 978-0025003408. Available also from: https://www.amazon.com/Guidebook-Learning-Lifelong-Pursuit-Wisdom/dp/0025003402?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0025003402.
3. AGARWAL, Rakesh. Fast Algorithms for Mining Association Rules. In: ***VLDB 1994***. 1994.
4. AGGARWAL, Charu C. ***Data Mining***. Springer International Publishing, 2015. isbn 3319141414. Available also from: https://www.ebook.de/de/product/23348463/charu_c_aggarwal_data_mining.html.
5. AGRAWAL, Rakesh; IMIELIŃSKI, Tomasz; SWAMI, Arun. Mining association rules between sets of items in large databases. ***ACM SIGMOD Record***. 1993, vol. 22, no. 2, pp. 207–216. Available from doi: 10.1145/170036.170072.
6. ANDREAS MEIER, Michael Kaufmann. ***SQL & NoSQL Databases***. Springer-Verlag GmbH, 2019. isbn 3658245484. Available also from: https://www.ebook.de/de/product/34744404/andreas_meier_michael_kaufmann_sql_nosql_databases.html.

7. BALDASSARRE, Michele. Think big: learning contexts, algorithms and data science. ***Research on Education and Media***. 2016, vol. 8, no. 2, pp. 69–83. Available from doi: 10.1515/rem-2016-0020.

8. BARTOLI, Alberto; DE LORENZO, Andrea; MEDVET, Eric; TARLAO, Fabiano. Inference of regular expressions for text extraction from examples. ***IEEE Transactions on Knowledge and Data Engineering***. 2016, vol. 28, no. 5, pp. 1217–1230.

9. BATINI, Carlo; SCANNAPIECO, Monica. ***Data Quality***. Springer-Verlag GmbH, 2006. isbn 3540331727. Available also from: https://www.ebook.de/de/product/5330024/carlo_batini_monica_scannapieco_data_quality.html.

10. BAWDEN, David; ROBINSON, Lyn. ***Introduction to Information Science (Foundations of the Information Sciences)***. Facet Publishing, 2012. isbn 978-1-85604-810-1. Available also from: https://www.amazon.com/Introduction-Information-Science-Foundations-Sciences/dp/1856048101?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1856048101.

11. BELLINGER, Gene; CASTRO, Durval; MILLS, Anthony. ***Data, Information, Knowledge, and Wisdom*** [online]. 2007-11-08 [visited on 2020-02-25]. Available from: http://systems-thinking.org/dikw/dikw.htm.

12. BRIDGWATER, Adrian. ***The 13 Types Of Data*** [online]. 2018-07-05 [visited on 2020-03-17]. Available from: https://www.forbes.com/sites/adrianbridgwater/2018/07/05/the-13-types-of-data/.

13. BRIN, Sergey; MOTWANI, Rajeev; ULLMAN, Jeffrey D.; TSUR, Shalom. Dynamic itemset counting and implication rules for market basket data. In: ***Proceedings of the 1997 ACM SIGMOD international conference on Management of data - SIGMOD '97***. ACM Press, 1997. Available from doi: 10.1145/253260.253325.

14. BROWNLEE, Jason. ***How Much Training Data is Required for Machine Learning?*** [Online]. 2017-07-24 [visited on 2020-07-09]. Available from: https://machinelearningmastery.com/much-training-data-required-machine-learning/.

15. BROWNLEE, Jason. ***A Gentle Introduction to LSTM Autoencoders*** [online]. 2018-11-05 [visited on 2020-07-09]. Available from: https://machinelearningmastery.com/lstm-autoencoders/.

16. CHARALABIDIS, Yannis; ZUIDERWIJK, Anneke; ALEXOPOULOS, Charalampos; JANSSEN, Marijn; LAMPOLTSHAMMER, Thomas; FERRO, Enrico. The Open Data Landscape. In: ***The World of Open Data***. Springer International Publishing, 2018, pp. 1–9. Available from doi: 10.1007/978-3-319-90850-2_1.

17. CHIEN, Melody; JAIN, Ankush. ***Magic Quadrant for Data Quality Tools*** [online]. 2019-03-27 [visited on 2020-03-23]. Tech. rep., ID G00363493. Gartner. Available from: https://www.gartner.com/en/documents/3905769.

18. CHRISTENSSON, Per. ***Data Definition*** [online]. 2006 [visited on 2020-02-27]. Available from: https://techterms.com/definition/data.

19. EHRLINGER, Lisa; RUSZ, Elisa; WÖß, Wolfram. A Survey of Data Quality Measurement and Monitoring Tools. ***arXiv preprint arXiv:1907.08138***. 2019.

20. ELSHAER, Ibrahim. ***What is the Meaning of Quality?*** 2012-05. MPRA Paper, 57345. University Library of Munich, Germany. Available also from: https://ideas.repec.org/p/pra/mprapa/57345.html.

21. FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. The KDD process for extracting useful knowledge from volumes of data. ***Communications of the ACM***. 1996, vol. 39, no. 11, pp. 27–34.

22. FLORIDI, Luciano. ***Philosophy and Computing***. Taylor & Francis Ltd, 1999. isbn 0415180244. Available also from: https://www.ebook.de/de/product/3646136/luciano_floridi_philosophy_and_computing.html.

23. FLORIDI, Luciano. ***The Philosophy of Information***. Oxford University Press(UK), 2013. isbn 0199232393. Available also from: https://www.ebook.de/de/product/19829123/luciano_floridi_the_philosophy_of_information.html.

24. FOUNDATION, Open Knowledge. ***What is Open Data?*** [Online]. 2019 [visited on 2020-03-03]. Available from: http://opendatahandbook.org/guide/en/what-is-open-data/.

25. GREENGARD, Samuel. ***10 Top Data Quality Tools*** [online]. 2019-06-20 [visited on 2020-03-25]. Available from: `https://www.datamation.com/big-data/10-top-data-quality-tools.html`.

26. HAHSLER, Michael. ***A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules*** [online]. 2020-05-14 [visited on 2020-07-20]. Available from: `https://michael.hahsler.net/research/association_rules/measures.html`.

27. HAHSLER, Michael; GRÜN, Bettina; HORNIK, Kurt. arules- A Computational Environment for Mining Association Rules and Frequent Item Sets. ***Journal of Statistical Software***. 2005, vol. 14, no. 15. Available from doi: `10.18637/jss.v014.i15`.

28. HAUSENBLAS, Michael. ***5 star data*** [online]. Ed. by KIM, James G. Boram. 2012-01-22 [visited on 2020-03-04]. Available from: `https://5stardata.info/en/`.

29. HEATH, Tom; BIZER, Christian. Linked Data: Evolving the Web into a Global Data Space. ***Synthesis Lectures on the Semantic Web: Theory and Technology***. 2011, vol. 1, no. 1, pp. 1–136. Available from doi: `10.2200/s00334ed1v01y201102wbe001`.

30. HOMAYOUNI, Hajar; GHOSH, Sudipto; RAY, Indrakshi. Data Warehouse Testing. In: ***Advances in Computers***. Elsevier, 2019, pp. 223–273. Available from doi: `10.1016/bs.adcom.2017.12.005`.

31. HUANG, Hong. Big data to knowledge – Harnessing semiotic relationships of data quality and skills in genome curation work. ***Journal of Information Science***. 2018, vol. 44, no. 6, pp. 785–801. Available from doi: `10.1177/0165551517748291`.

32. HULSEBOS, Madelon; HU, Kevin; BAKKER, Michiel; ZGRAGGEN, Emanuel; SATYANARAYAN, Arvind; KRASKA, Tim; DEMIRALP, Çagatay; HIDALGO, César. Sherlock. In: ***Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining***. ACM, 2019. Available from doi: `10.1145/3292500.3330993`.

33. INFORMATICA. ***Informatica (Version 9.6.0)***. 2014-01-01. Available also from: `https://kb.informatica.com/proddocs/Product%20Documentation/2/DQ_960_RuleBuilderGuide_en.pdf`.

34. JANZ, Danny. ***Data Exploration with Adversarial Autoencoders*** [online]. 2019-07-20 [visited on 2020-07-10]. Available from: https://towardsdatascience.com/data-exploration-with-adversarial-autoencoders-311a4e1f271b.

35. KERNER, Sean Michael. ***Translytical data platforms provide options to unify databases*** [online]. 2019-11-06 [visited on 2020-03-23]. Available from: https://searchdatamanagement.techtarget.com/news/252473595/Translytical-data-platforms-provide-options-to-unify-databases.

36. KERREMANS, Philippe. ***Hybrid Transactional/Analytical Processing (HTAP)*** [online]. 2018-05-04 [visited on 2020-03-23]. Available from: http://www.bigdatareflections.net/blog/?p=109.

37. KIRKLAND, James. ***Internet of Things: insights from Red Hat*** [online]. 2015-03-31 [visited on 2020-06-25]. Available from: https://developers.redhat.com/blog/2015/03/31/internet-of-things-insights-from-red-hat/.

38. KUMAR, Ashish; VILLAMARIONA, Jorge. ***Data Lake Essentials, Part 1 – Storage And Data Processing*** [online]. 2020-01-23 [visited on 2020-03-23]. Available from: https://www.qubole.com/blog/data-lake-essentials-part-1-storage-and-data-processing/.

39. KUMBHARE, Trupti A.; CHOBE, Santosh V. An Overview of Association Rule Mining Algorithms. In: 2014.

40. LIU, Xiaobing; ZHAI, Kun; PEDRYCZ, Witold. An improved association rules mining method. ***Expert Systems with Applications***. 2012, vol. 39, no. 1, pp. 1362–1374. Available from doi: 10.1016/j.eswa.2011.08.018.

41. MAHANTI, Rupa. ***Data quality : dimensions, measurement, strategy, management, and governance***. Milwaukee, Wisconsin: ASQ Quality Press, 2018. isbn 9780873899772.

42. MALIK, Usman. ***Association Rule Mining via Apriori Algorithm in Python*** [online]. 2018-08-09 [visited on 2020-07-18]. Available from: https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/.

43. AL-MAOLEGI, Mohammed; ARKOK, Bassam. An Improved Apriori Algorithm For Association Rules. ***International Journal on Natural Language Computing***. 2014, vol. 3, no. 1, pp. 21–29. Available from doi: 10.5121/ijnlc.2014.3103.

44. MCDANIEL, Stacey. ***How to Choose the Right Data Quality Tools*** [online]. 2019-06-17 [visited on 2020-04-07]. Available from: https://www.talend.com/resources/data-quality-tools/.

45. MCGILVRAY, Danette. ***Executing Data Quality Projects***. Elsevier LTD, Oxford, 2008. isbn 0123743699. Available also from: https://www.ebook.de/de/product/7336209/danette_mcgilvray_executing_data_quality_projects.html.

46. MORENO, María N; SEGRERA, Saddys; LÓPEZ, Vivian F; POLO, M José. Improving the quality of association rules by preprocessing numerical data. In: ***II Congreso Español de Informática***. 2007, pp. 223–230.

47. ORLANDO, Salvatore. ***What is Data?*** [Online]. 2015 [visited on 2020-03-10]. Available from: https://www.dais.unive.it/~dm/New_Slides/2_Data_DWM.pdf.

48. PEARLMAN, Shana. ***Data Lake vs Data Warehouse: What's the Difference?*** [Online]. 2019-10-10 [visited on 2020-03-23]. Available from: https://www.talend.com/resources/data-lake-vs-data-warehouse/.

49. PEARLMAN, Shana. ***Data Quality Tools – Why the Cloud is the Cure for Dirty Data*** [online]. 2019-10-23 [visited on 2020-04-07]. Available from: https://www.talend.com/resources/data-quality-tools-cloud/.

50. PIPINO, Leo L.; LEE, Yang W.; WANG, Richard Y. Data quality assessment. ***Communications of the ACM***. 2002, vol. 45, no. 4, p. 211. Available from doi: 10.1145/505248.506010.

51. PURKAIT, Niloy. ***Hands-On Neural Networks with Keras: Design and create neural networks using deep learning and artificial intelligence principles***. Packt Publishing Ltd, 2019.

52. ROCCA, Joseph; ROCCA, Baptiste. ***Understanding Variational Autoencoders (VAEs)*** [online]. 2019-09-24 [visited on 2020-07-05]. Available from: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

53. ROLLAND, Alexis. ***MobyDQ*** [online]. 2020-01-01 [visited on 2020-04-07]. Available from: https://ubisoftinc.github.io/mobydq/.

54. ROWLEY, Jennifer. The wisdom hierarchy: representations of the DIKW hierarchy. ***Journal of Information Science***. 2007, vol. 33, no. 2, pp. 163–180. Available from doi: 10.1177/0165551506070706.

55. SAP. ***We know Data. We know Platforms. Let's Talk Translytics.*** [Online]. 2018-01-03 [visited on 2020-03-23]. Available from: https://blogs.sap.com/2018/01/03/we-know-data-we-know-platforms-lets-talk-translytics/.

56. SEVERINI, Luca. ***DIKW Paradigm Today*** [online]. 2016 [visited on 2020-02-27]. Available from: https://www.slideshare.net/epistematica/dikw-paradigm-today.

57. SHANKS, Graeme; DARKE, Peta. Understanding data quality in a data warehouse: a semiotic approach. In: ***Proceedings of conference on information quality***. University of Massachusetts Lowell, 1998, pp. 292–309. Conference on Information Quality ; Conference date: 01-01-1998.

58. SMITH, Tom. ***Data DevOps - What and Why*** [online]. 2019-07-30 [visited on 2020-03-30]. Available from: https://dzone.com/articles/data-devops-what-and-why.

59. TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. ***Introduction to Data Mining***. PEL, 2013. isbn 978-1-292-02615-2. Available also from: https://www.amazon.com/Introduction-Data-Mining-Pang-Ning-Tan/dp/1292026154?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1292026154.

60. WANG, Richard Y.; STRONG, Diane M. Beyond accuracy: What data quality means to data consumers. ***Journal of management information systems***. 1996, vol. 12, no. 4, pp. 5–33.

61. WICKHAM, Hadley. Tidy data. ***The Journal of Statistical Software***. 2014, vol. 59. Available also from: http://www.jstatsoft.org/v59/i10/.

62. ZAKI, Mohammed J; MEIRA, Wagner. ***Data mining and analysis: fundamental concepts and algorithms***. Cambridge University Press, 2014.

63. ZELENY, Milan. Management support systems: Towards integrated knowledge management. ***Human Systems Management***. 1987, vol. 7, no. 1, pp. 59–70. issn 0167-2533. Available from doi: 10.3233/HSM-1987-7108.

64. ZHANG, Shichao. ***Association Rule Mining***. Springer Berlin Heidelberg, 2002. isbn 3540435336. Available also from: https://www.ebook.de/de/product/6701874/shichao_zhang_association_rule_mining.html.

65. ZINS, Chaim. Conceptual approaches for defining data, information, and knowledge. ***Journal of the American Society for Information Science and Technology***. 2007, vol. 58, no. 4, pp. 479–493. Available from doi: 10.1002/asi.20508.

Appendix **A**

# Supplement to Data quality

The following sections contain theoretical supplement that did not fit into the main content of the work.

## A.1 Data quality unit

For a full understanding of data quality methods it is practical to have an overview of the inputs and outputs of a general method. To illustrate and understand the context of the methods, a theoretical unit of data quality has been developed (see Figure A.1), which depicts data quality methods for higher abstraction. It is only a theoretical scheme intended to consolidate the data and application context of data quality methods. The schema provides an option for possible expansion. The scheme follows the general needs of data quality (see Section 1.3.1 about Selection criteria for data quality tool).

Data quality unit is a novelty idea that was invented based on the findings in this diploma thesis.

First we need to understand the incoming data (see Section 1 about Theoretical framework). The inputs to the data quality method are data categories (see Section 1.1.2.7 about Data categories) that should ideally contain the data itself. They may also contain metadata information about the input data (this is a form of Data profiling)(see Section 1.2 about Understanding data quality):

- data scheme (a structure for organizing and classifying data),

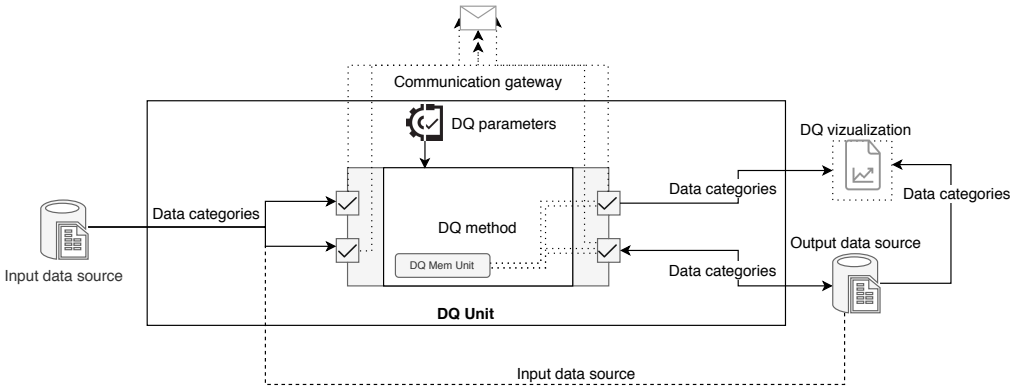


**Figure A.1:** Data quality unit.

- data format,
- data size,
- an expected time of arrival,
- an expected range of values.

The metadata mentioned above can be automatically determined from the data of DQ Mem Unit (Data Quality Memory Unit). DQ Mem Unit is a memory unit for storing metadata about its operation ideally in the short term (e.g. information about previous data within a time window).

Ideally, the input values of the method should be tested automatically. If any of the values do not comply with the set rules, for example in DQ parameters or DQ Mem Unit, an adequate message should be sent in a timely manner via the communication channel to the communication gateway (e.g. email, SMS or using a communication platform like Slack, Hangouts Chat, Microsoft Teams). For example, this may be an indication that the expected data format has changed or that the size of the input file does not match average values (e.g. Anomaly Detection). It is a form of data quality monitoring.

Furthermore, a data quality method (e.g. Data cleansing) itself is performed, that can be parameterized (DQ parameters – data requirements or data quality algorithm), as well. Method progress and data information (e.g. data profiling information) are recorded in DQ Mem Unit in a queryable structured data format and appropriately historized for monitoring purposes. These data may

engage long-term checks to monitor data quality trends (e.g. the percentage of recorded null values for the last week varies by more than two standard deviations for data for the last month or data change within a specified window). At the end of an ongoing method, its performance can also be evaluated (e.g. a slowdown in the method performance was detected within one week) and a message about performance issue should be sent.

In addition to the alarm messages mentioned above and to the storage of operational data, we may need to store the output from the data quality method in the output data source and perform adequate checks on the output data. For example, in the case of a data enrichment or data cleansing method – we need to store the data and check whether they meet the requirements (e.g. schema, format, referential integrity). The data may also be provided with data quality tags (e.g. information about a source of enrichment for a record), for example, in the form of metadata. Data quality output information can be visualized, as well. The read performance of Output data source should not be affected by Data Quality Unit.

Appendix B

# List of Acronyms

| | |
|---|---|
| **AAE** | Adversarial Autoencoder |
| **ACID** | Atomicity, Consistency, Isolation, Durability |
| **AI** | Artificial Intelligence |
| **BASE** | Basically Available, Soft-state, Eventually consistent |
| **BA** | Business Analytic |
| **BFS** | Breadth-first search |
| **BI** | Business Intelligence |
| **BOW** | Bag-of-Words |
| **CAP** | Consistency, Availability, Partition tolerance |
| **CEN** | The European Committee for Standardization |
| **CRS** | Compressed Row Storage |
| **DQ** | Data quality |
| **ELT** | Extract, Load, Transform |
| **ETL** | Extract, Transform, Load |
| **GDI** | General Definition of Information |
| **GIGO** | Garbage in, garbage out |
| **HTAP** | Hybrid transactional/analytical processing |
| **HTML** | Hypertext Markup Language |
| **HTTP** | Hypertext Transfer Protocol |
| **IoT** | Internet of things |
| **JSON** | JavaScript Object Notation |
| **KDD** | Knowledge Discovery in Databases |

| | |
|---|---|
| **LOD** | Linked Open Data |
| **MBA** | Market Basket Analysis |
| **MDM** | Master Data Management |
| **MIT** | Massachusetts Institute of Technology |
| **ML** | Machine Learning |
| **MSE** | Mean Squared Error |
| **NLP** | Natural language processing |
| **NLTK** | Natural Language Toolkit |
| **NoSQL** | Not only SQL |
| **OLAP** | Online Analytical Processing |
| **OLTP** | Online Transactional Processing |
| **ORC** | Optimized Row Compressed |
| **OWL** | The Web Ontology Language |
| **RAM** | Random-access memory |
| **RDFS** | RDF Schema |
| **RDF** | Resource Description Framework |
| **RTD** | Real-time data |
| **SKOS** | Simple Knowledge Organization System |
| **SSD** | Solid-state drive |
| **TF-IDF** | Term Frequency–Inverse Document Frequency |
| **TSV** | Tab Separated Value |
| **URI** | Uniform Resource Identifiers |
| **URL** | Uniform Resource Locator |
| **VAE** | Variational Autoencoder |
| **XML** | Extensible Markup Language |
| **ZIP** | Zone Improvement Plan |

Appendix C

# Supplemental Material

The thesis files and the complete experiments source codes, data and associated files can be found on the attached medium.

```
├── Thesis/ ..... XeLaTeX source codes and PDF of the thesis
└── Experiments/ ..... a repository with experiments
```

**Directory structure C.1:** Contents of the attached medium

Appendix D

# Appendix files list

The following list contains files that did not fit into the main content of this thesis. All files can be found in the repository .\Thesis\media\appendix_files*.

- DQ-tools-analysis.xlsx – The fundamental analysis of existing data quality tools.
- environment.yml – Conda environment file for the experiments.
- Experiment-1-autoencoder-macys-loss-columns.pdf – Autoencoder performance graphs (Macy's dataset).
- Experiment-1-autoencoder-macys-mse-dist-columns.pdf – Histograms - distribution of MSE for features (Macy's dataset).
- Experiment-1-autoencoder-macys-mse-summary-results.csv – Summary results for each feature (Macy's dataset).
- Experiment-1-autoencoder-foods-loss-columns.png – Autoencoder performance graphs (Open Food Facts dataset).
- Experiment-1-autoencoder-foods-mse-dist-columns.png – Histograms - distribution of MSE for features (Open Food Facts dataset).
- Experiment-1-autoencoder-foods-mse-summary-results.csv – Summary results for each feature (Open Food Facts dataset).