POLITECNICO DI TORINO

Master degree course in Data Science and Engineering

Master Degree Thesis

Quality analysis of the Italian open government data through a generalized algorithm

Supervisors
prof. Antonio Vetrò

Co-Supervisors
prof. Marco Torchiano

Candidates
Davide Vitaletti
matricola: 269462

Anno accademico 2020-2021

Abstract

The most important asset for today's economy is data. A lot of businesses, organizations and governments deeply analyze their data, from sales to health, in order to gain useful insights that can be exploited to improve services or products. Indeed, we are in the so-called data-driven economy. Lots of decisions are affected by data and many automated systems don't need any human supervision at all. As a result, having bad quality data will result in failures or unattended results in most cases. For this reason, having good data that is reliable is becoming a priority for many industries and public sectors. The aim of the thesis is to design and implement a generalized algorithm that can assess the data quality of a structured dataset following the data quality standard defined by ISO/IEC 25012 and ISO/IEC 25024. The algorithm is then exploited by assessing the quality of the Italian open government data. The assessment was conducted by considering several open datasets available in the official Italian site. The result is an analysis of the quality of the Italian open government data that highlights both strengths and weakness and suggests actions to improve the quality of the data.

Preface

Data is the new oil: information has always played an important role in making decisions and, in the current business world, data is the most precious asset. Data, however, is only useful if it's high-quality and, at the opposite, bad data can cause great loss for companies or organizations. For this reason, data quality is becoming increasingly important for business as data management techniques and technologies improve.

As result, it is important to assess the data quality of a file in order to highlight as soon as possible errors or anomalies; so the goal is to prevents errors rather than fix them later. The aim of this thesis, indeed, is to design and develop a generalized algorithm that can assess automatically the data quality of any CSV file without additional information.

In order to make an algorithm that can be globally acceptable, the method and the measures exploited to assess the data quality are taken or derived from the ISO 25012 and 25024, as described in chapter 2.

In chapter 3, instead, it is explained how those measures are technically implemented. Besides, the algorithm is exhaustively presented showing its input, its output, the core idea, how it works and some challenges that were faced during the implementation and testing.

Finally, in chapter 4, it is presented a practical application of the algorithm that is used to asses the data quality of the Italian open government data sets. The obtained results are described and analyzed in order to highlight criticalties and try to find the causes behind the anomalies. Also, considerations and suggestions are reported in order to improve the data quality of the data sets.

Acknowledgements

To my Girlfriend,
To my Family,
To my Friends,
To the Professors,
To Myself,

Thank you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction to data quality

1.1 The importance of data quality

In 2017, The Economist published an article [1] titled: *"The world's most valuable resource is no longer oil, but data"*. Since its publication, that topic has generated a great deal of discussion and "Data is the new oil" has become a common refrain.

In the current business world, data is one of the hottest topic. Many companies are interested in the insights and economic value they can derive from their data. Data is indeed one of the most valuable resources available to today's marketers, agencies, publishers, media companies, governments and more. Among the most important benefits of a data-driven approach that can be shared to any business, it can be selected: deeper understanding of its own market, improvements of marketing strategy, more accurate user personalizing, more reliable automation decision systems.

But data is only useful if it's high-quality and, at the opposite, bad data can cause great loss in terms of revenue and reputation. Also, if bad data or errors are not detected, they can lead to further contamination of the data sets. The direct economic damage caused by poor data quality issues could include additional costs associated with shipping products to the incorrect address, lost sales opportunities due to incorrect or incomplete customer records and improper financial or regulatory compliance reporting. IBM estimates that bad data costs the U.S. economy $3.1 trillion per year [2]. Those costs come mainly from the time employees have to spend correcting bad data

and errors that cause mistakes with customers and other businesses. The financial impact contributes to the growing trend of organizations seeking data quality solutions. Another study [3] pointed out that it's more convenient spending time and money in cleaning data record rather than enduring the cost of negative impact caused by bad data. The study reported that if the cost of preventing bad data is set to 1, the cost of correcting the impact of bad quality data is 10, while the cost of just accepting the negative impact is 100. The results are shown in figure 1.1

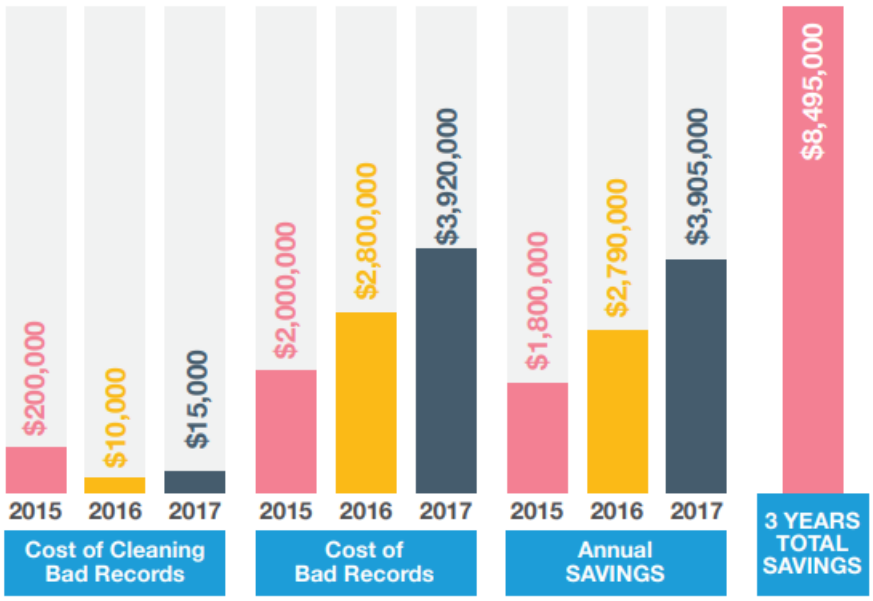

Figure 1.1. Savings in cleaning bad records rather than correcting the negative impact of them

Data quality is becoming increasingly important for businesses as data management techniques and technologies improve. Machine learning and, in general, automation technologies have enormous potential, but their success is heavily dependent on data quality. A predictive model, for example, need a large amount of accurate data to work properly. The more good data a machine learning algorithm has as input, the faster it can produce reliable results.

Rather than treating data as a separate function from the rest of their operations, today's most successful businesses incorporate it into everything they do. Because of this increased integration, data quality can have an impact on many aspects of a business, including marketing, sales, and content creation.

Data quality is also critical due to data compliance as GDPR. Data regulations are evolving dynamically so it's becoming increasingly important for businesses to properly manage their data. If data is disorganized or poorly maintained, it's more difficult to demonstrate compliance. This is especially important for financial and sensitive data, but it can also be applied to all types of data.

1.2 Data quality definition and models

Data quality is a metric that indicates how well a data set is suited to its intended use. Due to the definition, data quality is a relative concept that depends on the purpose. For this reason, the quality of a data set can be sufficient for certain goals and situations but inadequate for other purpose that requires other information. In order to fully assess the data quality from different angles, it's important to consider multiple dimensions. As stated from the study conducted from Richard Farnworth [4], data can be assessed in 6 dimensions:

- Accuracy: Refers to how well the data describes the real-world conditions it aims to describe. Inaccurate data lead to incorrect conclusions and these insights are not reliable and should not be exploited. For example, if a system for e-commerce management always inverts the gender of the user; from the data it can be shown that the majority of the clients are men though the reality is that the 80% of the users are women. So, that information is inaccurate.

- Completeness: it indicates how many information are present or not in the data set. In other words, it assess if everything that was supposed to be collected was successfully collected. Any data set may have gaps and missing data, but does that missing data is crucial for the business or the company? If a customer skipped several questions on a survey, for example, the data he submitted would not be complete but it is also important to asses how much relevant were the questions that were left blank.

- Consistency: refers to the uniformity of data as it moves across networks and applications. The same data values stored in difference locations should not conflict with one another. For a department store, as instance, data about a particular customer can be hold through a loyalty

program, mailing list, online accounts payment system and order fulfilment system. In that group of systems there may be misspelled names, old addresses and conflicting status flags.

- Validity: refers to the process of gathering data rather than the data itself. A field in a data set may have conditions which it needs to satisfy to be considered valid. If data is in the correct format and is of the correct type and falls within the correct range, it is considered as valid. For example, we can set that the date type information must be written as YYYY-MM-DD; all other date format are not valid in the data set.

- Relevancy: The data that were collected should be useful for the goals they were planed to be use for. Even if the information has all the other characteristics of quality data, if it's not relevant to the defined objective, it's not useful. For example, in a software that tracks down virus cases, it's not relevant having daily information about the weather.

- Timeliness: refers to how recently the event that the data represents occurred. Data should be recorded soon after the real world event. Data typically becomes less useful and less accurate as time goes on. This measures is fundamental for real-time decision making systems that require storing and processing real-time data. If the data from a radar system could only be downloaded in batches, once a day, it is not helpful and relevant to air-traffic controllers.

Chapter 2

ISO standards and measures

2.1 ISO/IEC 25012: model

The ISO/IEC 25012 has the aim of defining a general data quality model for computer systems, to support organizations to acquire, manipulate and use data with the necessary quality characteristics to achieve its objectives. As stated by the ISO itself, the model describes data quality characteristics for any computer system application so it must be general and worldwide applicable. The individual characteristic of the data quality model can vary in importance depending on specific requirements of the applications of the computer system. Also, the model considers all data types and rules that impact data structure and relationships between data (as consistency between data in the same or in different entities).

Data quality can be measured by the following model in which 18 characteristics of data quality attributes are taken into account: consistency, currency, completeness, precision, accuracy, security, availability, recoverability, understandability, manageability, efficiency, changeability, portability, productivity, safety, credibility, accessibility, regulatory compliance.

The Data Quality characteristics are classified in to main categories:

- Inherent Data Quality: refers to the degree to which quality characteristics of data have the intrinsic capacity to satisfy stated and implied needs.
- System-Dependent Data Quality: data quality depends on the technological domain in which data are used; data quality is achieved by the

capabilities of computer systems' components: hardware and software.

In figure 2.1 it is shown which data characteristics are inherent or system dependent

Characteristics	Data quality	
	Inherent	System dependent
Accuracy	X	
Completeness	X	
Consistency	X	
Credibility	X	
Currentness	X	
Accessibility	X	X
Compliance	X	X
Confidentiality	X	X
Efficiency	X	X
Precision	X	X
Traceability	X	X
Understandability	X	X
Availability		X
Portability		X
Recoverability		X

Figure 2.1. Segmentation of inherent data characteristics from system-dependent ones

2.1.1 Consistency

Consistency refers to the absence of apparent contradictions within data. Discrepancies among data can be spotted in the same data set: for example, the death date of a person can't be before the birth date. Data coherence can also be measured considering different data sets: for example, the total number of student attending first year in an university can't be greater than the total number of students attending that university.

2.1.2 Currency

Currency is the extent to which data is up-to-date and it can be measured with reference both to the reality and to the task to complete. For example:

if a customer booked a seat, the data about that seat should be updated as soon as possible. If not, another customer could book the same seat because data was not up-to-date.

2.1.3 Completeness

Completeness refers to how comprehensive the information is. In other words, it measures if all the needed information are presents in the data set. From an end-user perspective, completeness is the extent to which data are sufficiently able to satisfy users stated needs from quantitative point of view. So, for example, in an e-commerce system, the information about the customer's address should be present, otherwise the goods can't be sent.

2.1.4 Precision

Precision is the capability of the value assigned to an attribute to provide the degree of information needed in a specific context. For example, regarding numeric data, the precision is the number of decimals places to the right of the decimal point. Depending on the context, the system could need only few decimal to describe successfully the information (example: cost of a good) or more (example: design measures of an aircraft).

2.1.5 Accuracy

Accuracy is defined as the degree to which a data value conforms to its actual or specified value. Accuracy can be measured based on 2 different aspect:

- Syntactical accuracy: it is the degree to which the value in the data set is similar to a set of defined values in a domain that are considered syntactically correct. For example, common women name are: Mary, Patricia, Linda. Those names defined the domain. If a value is "Marj", the data is inaccurate from a syntactical point of view because it is in the right domain (women names) but spells incorrectly.

- Semantic accuracy: it is defined as the closeness of the data values to a set of values defined in a domain considered semantically correct. For example, the real name of a person is "Mary" but the stored name in the data set is "Linda". Each name is correct from a syntactical point of view because of the domain of reference in which they reside, but the information about the name is inaccurate.

2.1.6 Security

Security is the capability of the data to be accessed and interpreted only by authorized users. For instance: some information in a website are available only to authorized users.

2.1.7 Availability

Availability is the capability of data to be always retrievable. The concepts can be extended also for specific case as concurrent access (reading or updating data) and backups.

2.1.8 Recoverability

Although recoverability is often considered as a system-characteristic, the ISO focuses on data that must be recoverable. Indeed, recoverability is the data's ability to maintain and preserve a certain level of operations, as well as its physical and logical integrity, even in the event of failure. For example: if an IT system has a failure, the data stored in it should be recoverable.

2.1.9 Understandability

Understandability is about how easily the real meaning of data can be understood by users. For example: To represent a region , the standard acronym is more human-understandable than the corresponding numeric code. It is also important to point out that several additional information about the data in a data set are provided through metadata.

2.1.10 Manageability

It is the capability of data to be stored appropriately from a functional point of view. For example: data about revenue and costs should be stored as numeric in order to make algebraic operations on them.

2.1.11 Efficiency

Efficiency is the degree to which appropriate types and number of resources are provided in order to process efficiently data. For example, if we have 1 giga-byte of data, we don't need a 1 tera-byte database capacity.

2.1.12 Changeability

Changeability is the capacity of data to be modified in its type, length or assigned value for changes in technological environment, in requirements or in functional specifications. For example: if needed, the length of the address should be changeable.

2.1.13 Portability

It is the capability of data to be moved from one platform to another, while maintaining its existing quality. Portability is also referred as “Interoperability" and it means the degree to which data has attributes that allow for it to be applied in as many set of situations as possible.

2.1.14 Productivity

Productivity is the ability of data to enable users to complete tasks efficiently while using the least amount of resources possible, as well as the extent to which data is applicable and helpful for the task at hand. In other words, data are productive when they satisfy the user’s information needs.

2.1.15 Safety

The degree to which data are linked to their ability to reduce risk to people, businesses, properties, or the environment in a given context of use.

2.1.16 Credibility

Credibility is the extent to which data are regarded as true and credible by users. If data has been certified by an independent and trusted authority, it can be regarded credible. Credit risk data that has been approved by internal audit, for example, is regarded credible and can be used by banks to assess credit risk.

2.1.17 Accessibility

The ability to access data, especially by persons who require associative technology or customized configuration due to a disability.

2.1.18 Regulatory compliance

The capacity of data to adhere to current standards, conventions, or regulations, as well as other data quality norms in force. All credit card companies, for example, must be PCI compliant.

2.2 ISO/IEC 25024: measures

While ISO/IEC 25012 has the aim to define a set of dimensions that can used to asses the data quality, the ISO/IEC 25024 aims to define measures that can be used to quantify data characteristics. In other words, the data quality dimensions are measured by applying a measurement method that is a logical sequence of operations used to quantify properties with respect to a specific scale. The application of a measurement method to quantify a data quality dimension is called a QM. More than one QM can be used for the measurement of a data quality characteristics.

The QMs reported in the ISO/IEC 25024 can be used over the all data life cycle stages and for other process as data quality evaluation, support and implement IT services management processes, support improvement of data quality.

According to the scope of this work, only some QMs were considered among the ones presented in the ISO/IEC paper. The chosen QMs had to respect the following characteristic:

- relevant to quantify a data quality measurement
- independent of the data item type, format and characteristics in order to be generally applicable to all data sets
- computable by only considering the data itself without referring to dictionaries, external data sets and meta-data

Some of the QMs that didn't respect one of the previous characteristic were manipulated in order to retrieve other globally applicable measurements that were derived from the ISO/IEC 25024 original ones. The goal was to make those measurement applicable in a general algorithm. The derived measures will be marked as "DEV". The chosen QMs are now explained in details.

2.2.1 Com-I-1 DevA

The original QM is defined in the table 2.1

Id: COM-I-1 **Dimension**: Completeness **Name**: Record completeness	
Description	**Measurement function**
Completeness of data items of a record within a data file	X = A/B A = number of data items with associated value not null in a record B = number of data items of the record for which completeness can be measured

Table 2.1. Original COM-I-1 from ISO/IEC 25024

The ISO/IEC measure refers to the completeness of a single record and that information alone is not that helpful. In order to assess the completeness of the whole data set, the measure should be applied to the all records and the average should be taken as result. So, a derived QM is defined starting from the original one and it is described in table 2.2

Id: COM-I-1 DevA **Dimension**: Completeness **Name**: Data set completeness	
Description	**Measurement function**
Ratio of null values within a data file	Average of X where X = A/B A = number of null value in the whole data set B = number of data items considered

Table 2.2. COM-I-1 DevA derivative from ISO/IEC 25024

2.2.2 Com-I-5

The original QM is defined in the table 2.3

As stated in the ISO, the goal of this measure is to compute the percentage of the rows that are no completely non-null in the data set. The following QM is compliant with all the characteristics that had been defined earlier in the document and it can be exploited as is in a generalized algorithm.

Id: COM-I-5 **Dimension**: Completeness **Name**: Empty records in a data file	
Description	**Measurement function**
False completeness of records within a data file	X = 1-A/B A = number of records where all data items are empty B = number of records in a data file

Table 2.3. Original COM-I-5 from ISO/IEC 25024

2.2.3 Acc-I-4

The original QM is defined in the table 2.4

Id: COM-I-5 **Dimension**: Accuracy **Name**: Risk of data set inaccuracy	
Description	**Measurement function**
The number of outliers in values is indicating a risk of inaccuracy for data values in a data set	X = A/B A = number of data values that are outliers B = number of data values to be considered in a data set

Table 2.4. Original ACC-I-4 from ISO/IEC 25024

For each column, this QM is applied, then the average is taken as result to characterize the whole data set. As for the Com-i-5, also this QM can be exploited as well in the generalized algorithm because it respects all the 3 characteristics. Also, the ISO suggests 3 possible technique to calculate outliers. Those techniques depends on the data distribution: normal distribution, any distribution, non-parametric. The method that is used in the generalized algorithm and the technical details will be discussed in the chapter 3.

2.2.4 Con-I-2 DevB

The original QM is defined in the table 2.5

Id: CON-I-2 **Dimension**: Consistency **Name**: Data format consistency	
Description	**Measurement function**
Consistency of data format of the same data item	X = A/B A = number of data items where the format of all properties is consistent in different data files B = number of data items for which format consistency can be defined

Table 2.5. Original CON-I-2 from ISO/IEC 25024

The original QM is not suited in a generalized algorithm because it depends on the specific properties of data items. For example, if it is known that there is a column "date" in the dataset, it is possible to assess whether all the data items in that attribute have the same date-format as YYYY-MM-DD or not. Instead, in a generalized algorithm, there are no prior information so it is impossible to know which properties the data itself should have. For this reason, a new measures that is general is derived and it is presented in the table 2.6

Id: CON-I-2 DevB **Dimension**: Consistency **Name**: Data type consistency	
Description	**Measurement function**
Average consistency of data type of data item in the same attribute	Average of X where X = A/B A = number of data items that have the correct type in the attribute B = number of data items considered for a single column

Table 2.6. CON-I-2 DevB derivative from ISO/IEC 25024

So, for each column, the percentage of data items that have the right type is computed; then the means of all those values is taken in order to have just one measure that described the whole data set. Lastly, is important to point out that the information about the type of the data item can be

retrieve by the algorithm itself without further information as metadata. For example: “home” is seen as string, while “4.32” is considered as a float. The correctness of the type in a column and the technical details will be discussed in the chapter 3.

2.2.5 Con-I-3

The original QM is defined in the table 2.7

Id: CON-I-3 **Dimension**: Consistency **Name**: Risk of data inconsistency	
Description	**Measurement function**
Risk of having inconsistency due to duplication of data value	X = A/B A = Number of data items where exist duplication in value B = Number of data items considered

Table 2.7. Original CON-I-3 from ISO/IEC 25024

This QM is suited as is and, as stated by the ISO, the duplication values can be found for each column but also grouping by k attributes then finding duplicates over the records/rows. In this scenario, duplication occurs when, for the set of k attributes selected, two or more records/rows are found equal. The actual implementation and further technical details will be discussed in chapter 3.

2.2.6 Con-I-4 DevC

The original QM is defined in the table 2.8

The QM refers to the data architecture in a broader view that include, also, the conceptual data model. This QM can be altered considering the architecture merely as synonymous of data structure, i.e. whether each row of the data set contains the expected number of value. For example, in a .csv file, if the header contains 4 column, each row should contain 4 values. As a result, the derived measures in defined in the following table 2.9

Id: CON-I-4 **Dimension**: Consistency **Name**: Architecture consistency	
Description	**Measurement function**
Degree to which the elements of the architecture have a correspondence in referenced architecture elements	X = A/B A = Number of elements of an architecture that have a corresponding referenced elements in the installed architecture B = Number of elements of the referenced architecture

Table 2.8. Original CON-I-4 from ISO/IEC 25024

Id: CON-I-4 DevC **Dimension**: Consistency **Name**: Data structure consistency	
Description	**Measurement function**
Degree to which the data structure remains coherent over the data file	X = A/B A = Number of rows that respect the data structure B = Number of rows contained in the data file

Table 2.9. Derivative CON-I-4 DevC from ISO/IEC 25024

Chapter 3

Study design

3.1 Core idea: a generalised algorithm

As stated in the previous chapters, the core idea of this work is to develop a generalised algorithm that can assess the data quality of any data set. The fist issue to tackle is that there are several data file format but the most popular are csv, json and xml. An example of those file format is presented in figure 3.1 and those format will be briefly explained as described by Susanne Morris [5]:

- CSV: is a data storage format that stands for Comma Separated Values. CSV files store data values in a list format separated by commas or other separation mark as semicolon or tab.

- JSON: is a data exchange format that stands for JavaScript Object Notation and it is known as a light-weight format type and is favored for its readability and nesting features.

- XML: stands for eXtensible Markup Language and it has a process for annotating data in a syntactically significant way using tab.

While CSV file are generally easy to manipulate even in cases of errors or missing values, the JSON and XML are are more difficult to handle because their structure is flexible. Indeed, there are several ways to set up a JSON file: each line could contain the whole record or just the single value; there could be several nesting in the same element; the single information could be store in an array and so on. In figure 3.2 it is shown how the same file could be write in 3 different JSON mapping.

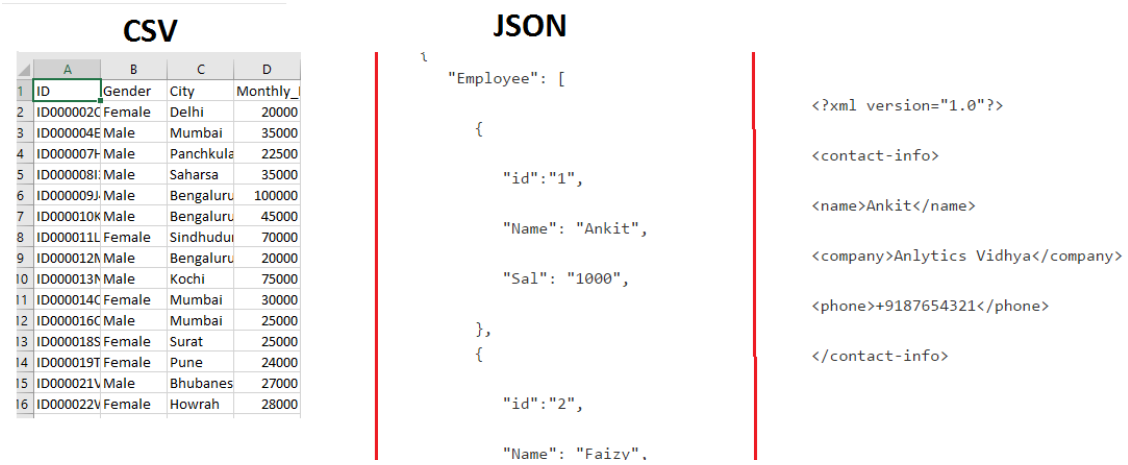

Figure 3.1. From left to right an example of a CSV, JSON and XML file

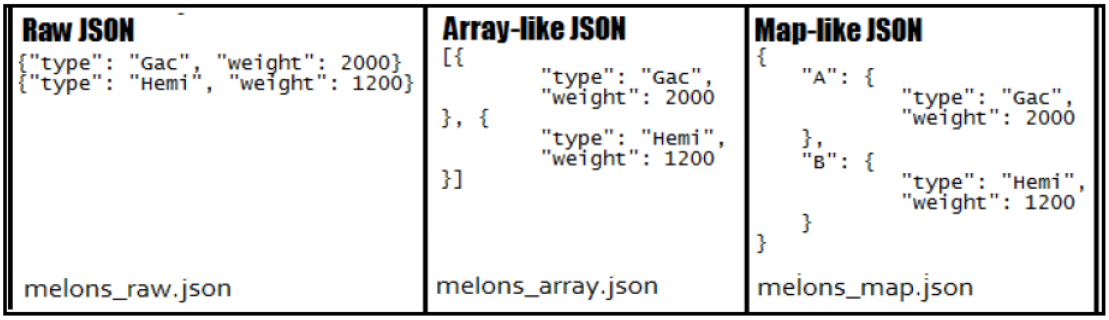

Figure 3.2. How the same information can be store in different JSON mapping

The first step in order to assess the data quality is to automatically retrieve all the data in the file; but for JSON, as it was shown, this can be difficult due to the flexibility of the mapping. It is also important to point out that the ideal algorithm is completely autonomous and independent, so no human actions or other information are required.

Therefore, in order to assess the JSON data quality, another algorithm that can define the type of mapping among all the possible existing mapping is required.

The same issue was found in dealing with XML file because also for this format there exist different mapping and every file could have unique tags. As a result, another algorithm is required to automatically understand the mapping of any XML and retrieve all the information.

Due to complexity and time limit, the proposed algorithm will only be suitable for CSV file. Indeed, from any CSV is it possible to create a data frame object that contains all the information in a table than can be easily manipulated and analyzed.

3.2 Input and output

The input of the program is a file (of format .csv or .txt) that contains in each row the URL or the path of a single csv data file. For each record, so for each csv, the program converts it into a dataframe and assesses its data quality computing the 6 measures that were define earlier: COM-I-5, ACC-I-4, CON-I-3, COM-I-1 DevA, Con-I-2 DevB, Con-I-4 DevC. The actual implementation of those measures will be discussed later. An example of an input file is reported in figure 3.3.

```
Comune Bologna
https://opendata.comune.bologna.it/api/v2/catalog/datasets/siepi/exports/csv
https://opendata.comune.bologna.it/api/v2/catalog/datasets/origini-di-bologna-strade/exports/csv
https://opendata.comune.bologna.it/api/v2/catalog/datasets/elezioni-amministrative-2016-preferenze-lista-cittadini-per-bologna/exports/csv
https://opendata.comune.bologna.it/api/v2/catalog/datasets/bologna-rilevazione-airbnb/exports/csv
https://opendata.comune.bologna.it/api/v2/catalog/datasets/centri-estivi-2020-posizione-geografica/exports/csv
https://opendata.comune.bologna.it/api/v2/catalog/datasets/principali-nazionalita-residenti-stranieri-per-quartiere/exports/csv
https://opendata.comune.bologna.it/api/v2/catalog/datasets/spese-del-consiglio-comunale/exports/csv
```

Figure 3.3. Example of input data that contains 7 data file from the municipality of Bologna

As a result, for each csv a specific record is created in the output file that contains all the data quality measures computed on that csv. Also, other 2 attributes are computed for each data file: the first is a boolean that indicates whether the file was successfully opened or nor; the second, instead, highlights the error generated when the csv can't be opened. Also, in case of error, the quality measures can't be computed so all value for that specific record will be "NaN". The output file generated from the example input file is shown in figure 3.4. As we can see, the attributes of the output file are the following:

- Link: URL or path of the single CSV file
- Can open: whether the file was successfully opened (1) or not (0)
- Com-I-1 DevA: data quality measure about completeness derived from the Com-I-1 of the ISO

- Com-I-5: Data quality about completeness taken from the ISO
- Acc-I-4; Data quality about accuracy taken from the ISO
- Con-I-3: Data quality about consistency taken from the ISO
- Con-I-3 DevB: data quality measure about consistency derived from the Con-I-3 of the ISO
- Con-I-4 DevC: data quality measure about consistency derived from the Con-I-4 of the ISO
- Error: describe the error returned by the algorithm when the file can't be opened

Link	Can Open	Com-I-1-D	Com-I-5	Acc-I-4	Con-I-3	Con-I-2-D	Con-I-4-D	Error
https://op	1	0.04	1.0	0.081	0.6	0.0	1.0	No error
https://op	1	0.0	1.0	nan	0.936	0.0	1.0	No error
https://op	1	0.0	1.0	0.045	2.357	0.0	1.0	No error
https://op	1	0.096	1.0	0.059	0.748	0.001	0.987	No error
https://op	1	0.0	1.0	0.0	0.655	0.001	1.0	No error
https://op	1	0.0	1.0	0.025	0.8	0.0	1.0	No error
https://op	1	0.0	1.0	0.048	2.459	0.001	1.0	No error

Figure 3.4. Output file generated by analyzing the data quality of 7 data file from the municipality of Bologna

3.3 Measures implementation

3.3.1 COM-I-5

This QM computes the percentage of the rows over the data set that are not completely non-null. To do so, the following simple function is used.

Listing 3.1. Com-I-5 Implementation

```
i=0
for el in df.isnull().all(1).values:
    if (el):
        i+=1
ratio = round(1-i/n_rows,n_of_decimal)
```

The code "df.isnull().all(1).values" maps every rows of the data frame to "true" only if all the values of that record are "NaN'; then it return a list that contains the value "true' or "false" for every row.

3.3.2 ACC-I-4

The QM computes the ratio of outliers in the whole data set; so, for every numeric column, the function stores the number of outliers and the number of considered elements. Then it sum up all the outliers and all the considered elements, it takes the division and the measure is computed. The key point of this QM is to find an appropriate outlier detection method. Indeed, the ideal model should work fine for all kind of data distribution and it shouldn't require any fine tuning to optimize the hyper parameters, due to time constrains (a data set could contains 1 million of rows and several columns). So, the model that was exploited was the IQR, as suggested by the ISO/IEC itself. Following [6], the method is briefly explained. So, for each attribute:

- Convert it to a list of numbers
- Find the first quartile, Q_1
- Find the third quartile, Q_3
- Calculate the $IQR = Q_3 - Q_1$
- Define the range as: $[Q_1 - 1.5 * IQR; Q_3 + 1.5 * IQR]$
- Any data point of the attribute outside this range is considered as potential outlier

Another method that was initially used was the DBSCAN clustering with variable hyperpameters set as:

- $eps = mean/2$ where $mean$ is the mean of the data list
- $min\_samples = len(x)/5$ where x is the data list

Although those values worked fine for the majority of the data set, it returned wrong outliers in dealing with data values too much big or too much small because the hyperparameters weren't appropriate for those conditions.

3.3.3 CON-I-3

This QM is used to find the ratio of duplication values over the data set. Duplication values can be found for each attributes over the csv table but they can also be found grouping by k attributes. So duplication occur also when, for the set of k attributes selected, two or more records are found equal. The all possible set of combination for a fixed k is $\begin{pmatrix} n \\ k \end{pmatrix}$ where n is the number of attribute and $k \in [1, n]$

Even if k can be any number up to n, the computations with $k > 2$ could be really heavy to process. Indeed, in some data sets n could be large ($n > 10$) or the number of records could be high as greater than 100000. Indeed, in the case of large data set, the computation have taken roughly 1 hour to complete just with $k = 2$; so with $k = 3$, several hours could be needed. Another problem is that, from a technical point of view, the function need as many for loop as the number of attributes considered together (k). But, due to the fact that k is dynamic, also the function should be built with dynamic structures where for loops are generated on the fly and this solution is not trivial. So, due to time and space constrains and technical difficulty, we have decided to consider only the cases with $k = 1$ (only one attribute is considered) and $k = 2$ (2 attributes grouped together).

So the total number of duplicates is computes as: $\sum_{k=1}^{2} \sum_{i=1}^{n} D_{i,k}$ where $D_{i,k}$ is the number of duplications found in set i of k-attributes. Finally, the denominator is computed as the total number of cells in the csv table. So, only in the case with $k = 1$ this QM return a values that is within 0 and 1; instead with $k > 2$, the measure can be greater than 1.

3.3.4 COM-I-1 DevA

This QM is about the ratio of null cells in the whole data set. The computation is trivial: the numerator is the number of null values in the data sets, while the denominator is the total number of cells. The following is the line of code that find the number of null values:

Listing 3.2. How to find the number of null values

```
n_nan = df.isna().sum().sum()
```

3.3.5 CON-I-2 DevB

This measure is about computing the ratio of type data inconsistency among the data set. In this study only 2 type were considered: numeric or string. So, firstly, for the column j, all the values are mapped as numeric or not. The mapping method is defined by the function ".isdigit()".

Then, if the majority of the values are numeric, then the attribute is considered as numeric and all the string values are considered as "error" from a consistency point of view; the opposite strategy is applied if the majority of the values are string. So, the data itself is used to determine the column type, therefore no prior knowledge about the attribute is required. Lastly, the algorithm computes the number of "error type" for every column and it sums all those values.

Finally, the ratio is computed considering this sum as the numerator and the number of cells considered as denominator. So, summing, up:

$$QM = \frac{\sum_{i=0}^{n} E_i}{\sum_{i=0}^{n} D_i} \tag{3.1}$$

Where E_i is the number of "error type" in column i, while D_i is the number of data items considered in column i

3.3.6 CON-I-4 DevC

The QM is about the architectural consistency of the data set, so it computes the ratio of the rows that have the "right" number of cells over the entire file. Firstly, there must be fixed a method to define the 'right" number of cells. Indeed, the algorithm defines as "right" the number of cells seen in the first row that is, often, the header. So, every row that have a larger or smaller number of cells is considered as "error" for the purpose of this computation. Indeed, the measure is just the ratio of the number of "right" rows over the total number of rows.

It is important to point out that, in this scenario, the counting of the number of cells for each line is performed by the algorithm itself using the method ".split("d")" where d is the delimiter. Them major challenge that was faced during the implementation was the following: in several data set, some cells contained a string value that were contained between double quotes. In those cases, if the delimiter is found inside a double quotes string, the information should not be divided because the cell is just one. For example a data set could have 3 attributes (name, surname, description) and a sample record could be:

Mario, Rossi, "Mario Rossi is a small man, he has studied philosophy at the university, he likes playing videogames"

The record is composed by 3 cells and every sophisticated algorithm (as "read_csv()" that is used to convert the csv to a dataframe) can handle quote. Instead, if a simple ".split(",")" method is used, the algorithm would see 5 cells because it would found 4 ",".

So a cleaning function was defined and it works as following for every line:

- The record is scanned and the algorithm saves the positions or indexes of the quotes symbol
- if there are no quote, the function is stopped because there is no need to clear the line; otherwise the function return a new line that is generated replacing the record within the quotes with " a "

Listing 3.3. Cleaning out the quotes from a record

```
#arr is the array containing the indexes of the quotes
#line is the original record
for i in range(0,len(arr)):
    if (i==0):
        new_line += line[:arr[i]]+'' a ''
        #the original record is taken until the
        #first quote
    elif (i==(len(arr)-1)):
        new_line += line[arr[i]:]
        #the original record is taken from the
        #last quote until the end
    elif i%2 != 0:
        new_line += line[arr[i]:arr[i+1]]+'' a ''
        #when i is uneven, we ignore the
        #line[arr[i]:arr[i+1]] because
        #it contains the record within the quotes
        #Instead, if i is even,
        #line[arr[i]:arr[i+1]] is outside of
        #the quotes so we can take it
return new_line
```

So, if this method is applied to the previous record, the *new_line* returned is:

Mario, Rossi, “ a ”

Then, applying the “.split(",")” method, the returned value is 3 that is the right number of cells.

3.4 Challenges

3.4.1 Determine the delimiter

The method that convert the csv to a dataframe is “read_csv” that has a parameter “delimiter” that, if specified by user, indicates the character to use to divided the row in cells. If it is not specified (sep=“None”), the standard methods tries to understand alone the delimiter of the csv. It works fine most of the time but, in some cases, the method can’t detect propery the delimiter. Indeed, if a csv is composed by records as the following when there is a field “description” but with no quotes; it may not work as expected.

Mario; Rossi; 25 years old; small, student, basket
Giorgio; Neri; 22 years old; tall, student, football

The standard method, as stated by several attempts, tends to favor the character “,” as delimiter; indeed it is the default one. So, in this scenario, the chosen delimiter will be “,” instead of “;” that is the correct one. This behaviour was indeed detected during the test phase dealing with the file [8].

To tackle these situations, a new function that has the goal of detecting the delimiter is implemented:

- The first row, that is likely to be the header, is selected
- The function finds the number of cells in that row using 3 different character as delimiter: “;”, “,”, tab
- The character that maximize the number of cells is the right one

Although this method is simple, it worked fine in all the data file used in Chapter 4.

3.4.2 Null values

When a cell is empty, the data item corresponds to “” so the value inside the cell is called “null” or “nan” (not a number). During the implementation, we

have extended the definition of “null” also to cells that contains as value “ ”. So the “ ” values are converted to “null” using this line of code:

Listing 3.4. Replacing a cell with only a white space with NaN

```
df.replace(r'^\s*$', np.NaN, regex=True)
```

Another critical point about null values is that, they are not considered when the following QM are computed: Acc-i-4, Con-i-2 DevB, Con-i-3,

3.4.3 Decoding and encoding UTF-8

The standard encoding for csv file is UTF-8; it is also the default encoding in the “read_csv()” method. But, in a few data set, there were problems with 1 ore more characters that were not supported by UTF-8. For this reason, in case of encoding error during the “read_csv()” method, the method is applied again with the following specification “read_csv(.., encoding='latin1')”. With that encoding, the characters not supported by UTF-8 can now be processed and no fatal error is raised.

A similar process is run when the first line of the csv file is read in order to detect the delimiter. In this case the line is firstly decoded with “latin” as following:

Listing 3.5. Line decoding with latin1

```
decoded_line = line.decode("latin1")
```

Also, the same line is used when the QM Con-i-4 DevC is run and an encoding error is raised.

3.4.4 HTTP Error 403: Forbidden

During the test, several network errors were found. THe most common one is the HTTP Error 404 but there is nothing that can be done in that case. Instead, there were also some HTTP Error 403 Forbidden that happens [7] when the user that is doing the request is not allowed to access. That happened for the data files of the region Basilicata and for the data file of the Ministry of Helath.

The key part is that if the request is made via Python script by Google COLAB, the 403 error is raised. Instead, if the request is made via browser by the user himself, the site can be reached and the file is downloaded. The solution is to fake the user-agent that is making the request: so, if Python is trying to open the URL, from the fake header of the request it will be as if

the browser, instead, is making the request. The code of lines that fake the user-agent information are the following:

Listing 3.6. Alter the user-agent information

```
req = Request(a)
req.add_header('User-Agent', 'Mozilla/5.0 (Windows NT
10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/94.0.4606.81 Safari/537.36')
newlink = urlopen(req)
```

The variable "newlink" include the original URL plus the altered information about the user-agent that is requesting that URL. Those lines are used in the following parts of the algorithm if a net error is raised:

- when converting the csv file to a dataframe
- when detecting the edelimiter in the csv file
- when computing QM Con-i-4 DevC

This solution worked fine for the data set of the Ministry of Health: before none of the URL were accessible from Python itself. Instead, about the data files of the region Basilicata, some of them were accessible using the fake user-agent, but others no and the error 403 were raised again. Due to time limit and the limited number of cases, a further investigation was not conducted in this work.

3.4.5 Empty dataframe

In other cases, the URL was valid and the data file was successfully converted to the data frame but the table was empty. In this scenario, no computation can be done so, in order to avoid errors, the following condition should be met by the data frame.

Listing 3.7. Detecting an empty dataframe

```
df.shape[0] == 0 #df.shape[0] is the number of rows
```

If the return value is True, so the data frame is empty, the algorithm is stopped and all the measures will be null.

3.4.6 Other errors

During the tests, other errors were handled as:

- HTTP 404 Not found: the URL is not valid, neither by python neither by browser.
- Cannot open: the data file can't be open because it is not a valid csv. Indeed, in some cases, the URL points to zip file or to a file without a format
- JSON respons: especially for the datasets of the region Toscana, the URL directs to a page that contains a JSON message that is a server bad response.

Chapter 4

Data quality in the italian ODG scenario

4.1 Introduction to OGD

In order to fully test the implemented algorithm, tons of different data sets are needed. Those data sets should also be open and globally available. So, as result, the algorithm is tested using the Italian open government data as known as OGD. Those data sets are available in the official site owned by AGID (Agenzia per l'Italia Digitale): https://www.dati.gov.it/

In the site, the data sets are grouped by public authorities that have published the data (municipalities, regions and public authorities as ministries). Also, at each data set it can be associated one or more tag that represents the thematic category (energy, environment, health, population, ...). Finally, it is possible to search group of data sets by selecting the public authority, the theme and the file format (csv, json, xml, ...) as shown in figure 4.1.

4.2 Evaluation of the OGD quality

The work is conducted by considering multiple data files belonging to the same authority. Indeed, for each authority, 25 data files are randomly selected and the quality is evaluated for each CSV.

This output can be used to make a deep analysis of the data quality but it is also important to have a general overview of the quality of the Italian OGD. For this reason, for each authority it is also computed the mean, max and min of each data quality attributes. All the general outputs are stored in

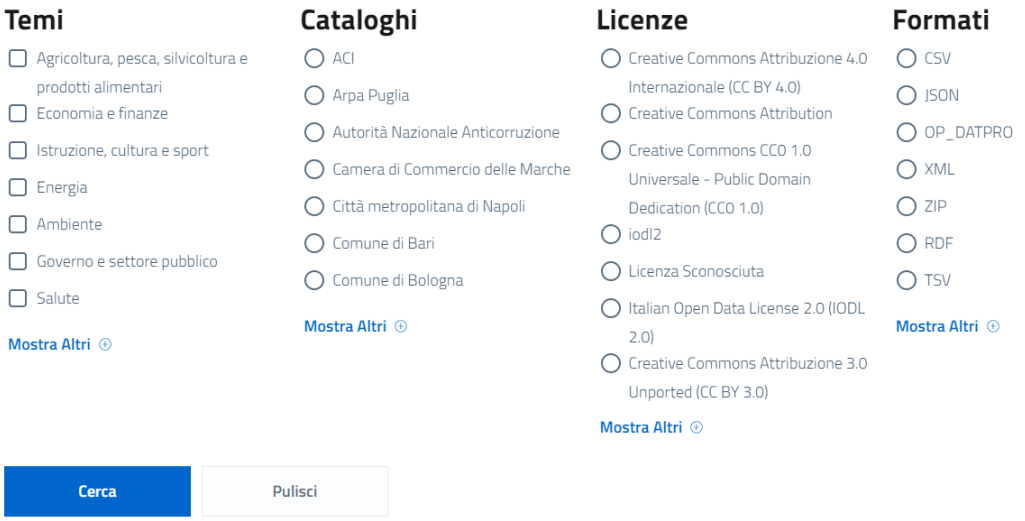

Figure 4.1. How Italian open government data sets can be found by filtering

3 different matrix: one for mean, max and min. In general, the final output file is a table where the attributes are the data quality measurements; the rows are the public authorities; the cells are the mean or max or min of that QM for that public authority as shown in figure 4.2, 4.3, 4.4. In those matrix, if the name attribute is green it means that greater values are better (Can open, Com-I-5, Con-I-4 DevC); while blue name attribute means that smaller values are better for that measurement (Com-I-1 DevA, Acc-I-4, Con-I-3, Con-I-2 DevB).

Among with the measures, also the information about the errors in opening the file were stored. Indeed, it is possible to see that some errors are more popular than other. All this information is stored in the image 4.5 where for each error it is shown how many times it has occur over the tests.

Ente	Can Open	Com-I-1-DevA	Com-I-5	Acc-I-4	Con-I-3	Con-I-2-DevB	Con-I-4-DevC
Comune Bologna	1	0,136	1	0,053	2,782	0,003	1
Comune Genova	0,967	0,05	0,974	0,049	0,615	0,004	0,951
Comune Lecce	0,967	0,076	1	0,085	2,372	0,018	0,976
Comune Milano	1	0,03	0,999	0,052	1,291	0,006	0,99
Comune Torino	0,933	0,04	1	0,042	4,044	0,01	1
INPS	1	0,011	0,998	0,051	0,644	0,013	1
Marche	0,85	0,03	1	0,036	0,588	0	0,995
MIBACT	1	0,005	1	0,088	0,901	0,01	1
Ministero dei trasporti	1	0,042	0,993	0,082	1,154	0,018	0,979
Ministero del Lavoro	1	0,062	1	0,038	0,723	0,029	1
ministero_salute	0,6	0,035	1	0,058	0,886	0,008	0,985
Miur	1	0,005	1	0,033	0,97	0,006	1
Regione Basilicata	0,633	0,032	1	0,052	0,825	0,017	0,947
Regione Campania	1	0,135	0,995	0,052	0,881	0,008	0,989
Regione EmiliaRomagna	0,7	0,157	0,9	0,032	0,79	0,009	0,939
Regione Friuli	1	0,06	1	0,069	1,913	0,004	0,998
Regione Lazio	0						
Regione Lombardia	1	0,079	1	0,056	0,841	0,008	0,995
Regione Piemonte	1	0,066	1	0,056	1,892	0,008	0,987
Regione Sardegna	0,8	0,07	1	0,046	0,784	0,005	0,956
Regione Sicilia	0,967	0,118	1	0,059	0,883	0,015	0,779
Regione Toscana	0,9	0,074	0,999	0,064	18,137	0,015	0,95
Regione Umbria	0,933	0,094	0,993	0,078	0,976	0,011	1
Regione Veneto	0,9	0,229	0,988	0,076	0,76	0,008	1
Average	0,881	0,071	0,993	0,057	1,985	0,01	0,975

Figure 4.2. Mean of data quality measurements for each authority

Ente	Com-I-1-DevA	Com-I-5	Acc-I-4	Con-I-3	Con-I-2-DevB	Con-I-4-DevC
Comune Bologna	0,621	1	0,308	57,701	0,035	1
Comune Genova	0,739	1	0,149	3,263	0,026	1
Comune Lecce	0,583	1	0,182	21,752	0,177	1
Comune Milano	0,205	1	0,225	7,133	0,054	1
Comune Torino	0,375	1	0,15	95,403	0,12	1
INPS	0,207	1	0,168	1,539	0,161	1
Marche	0,252	1	0,125	5,103	0	1
MIBACT	0,015	1	0,376	0,959	0,047	1
Ministero dei trasporti	0,215	1	0,3	8,037	0,125	1
Ministero del Lavoro	0,333	1	0,167	1,041	0,083	1
ministero_salute	0,336	1	0,17	2,318	0,057	1
Miur	0,074	1	0,099	1,438	0,054	1
Regione Basilicata	0,202	1	0,149	3,978	0,153	1
Regione Campania	0,501	1	0,165	4,33	0,061	1
Regione EmiliaRomagna	0,592	1	0,111	5,737	0,129	1
Regione Friuli	0,597	1	0,178	21,539	0,05	1
Regione Lazio						
Regione Lombardia	0,444	1	0,178	2,364	0,13	1
Regione Piemonte	0,379	1	0,246	35,959	0,125	1
Regione Sardegna	0,238	1	0,112	1,286	0,034	1
Regione Sicilia	0,657	1	0,158	6,15	0,076	1
Regione Toscana	0,949	1	0,227	475,37	0,091	1
Regione Umbria	0,555	1	0,215	7,981	0,101	1
Regione Veneto	0,5	1	0,193	3,586	0,042	1

Figure 4.3. Max of data quality measurements for each authority

Ente	Com-I-1-DevA	Com-I-5	Acc-I-4	Con-I-3	Con-I-2-DevB	Con-I-4-DevC
Comune Bologna	0	1	0	0	0	0,987
Comune Genova	0	0,261	0	0	0	0
Comune Lecce	0	0,99	0	0	0	0,429
Comune Milano	0	0,971	0	0	0	0,737
Comune Torino	0	1	0	0	0	1
INPS	0	0,947	0	0,012	0	1
Marche	0	1	0	0	0	0,914
MIBACT	0	1	0	0,58	0	1
Ministero dei trasporti	0	0,8	0	0	0	0,533
Ministero del Lavoro	0	1	0	0	0	1
ministero_salute	0	1	0	0	0	0,873
Miur	0	1	0	0,745	0	1
Regione Basilicata	0	1	0	0	0	0
Regione Campania	0	0,857	0	0	0	0,857
Regione EmiliaRomagna	0	0,5	0	0	0	0,168
Regione Friuli_	0	1	0	0,324	0	0,952
Regione Lazio						
Regione Lombardia	0	1	0	0	0	0,9
Regione Piemonte	0	1	0	0	0	0,625
Regione Sardegna	0	1	0	0	0	0,011
Regione Sicilia	0	0,99	0	0,243	0	0
Regione Toscana	0	0,987	0	0	0	0,048
Regione Umbria	0	0,882	0	0,067	0	1
Regione Veneto	0	0,667	0	0	0	1

Figure 4.4. Min of data quality measurements for each authority

Error	Occurence
HTTP Error 404: Not Found	6
cannot unpack non-iterable NoneType object	5
HTTP Error 503: Service Unavailable	3
This file is a zip	12
HTTP Error 403: Forbidden	8
<class 'urllib.error.URLError'>	43
The table has no rows	2
HTTP Error 400: Bad Request	1
File starts with a {, it is a json response message	3
<class 'pandas.errors.EmptyDataError'>	1

Figure 4.5. The number of occurrences for each error

4.3 Analysis of the results

4.3.1 Opening files and errors

This measure represents the ratio of the csv files that were successfully opened and analyzed. If the average is computed considering all the public authorities, the result is that 88,1% of csv were opened as shown in figure

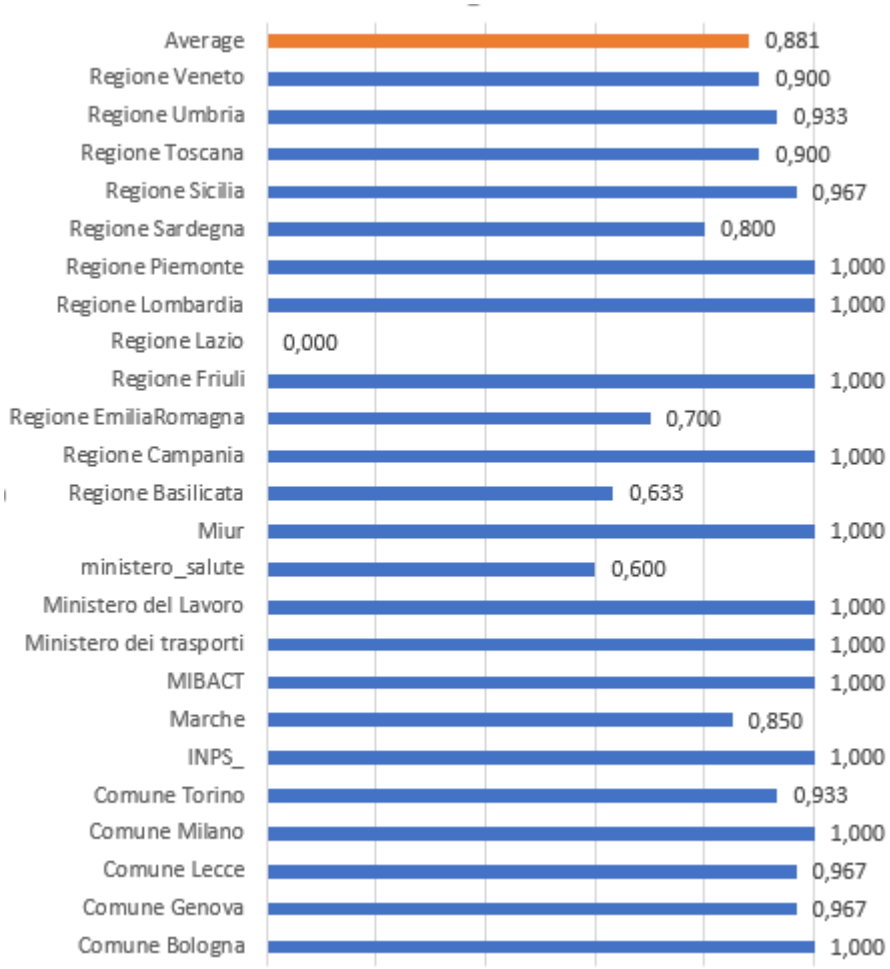

Figure 4.6. Visualization of the percentage of files that were successfully opened

If the data about the region Lazio is not considered, the ratio reaches 0,92% that is a good percentage. Indeed, the dataset about the Lazio region deserves a few consideration. As it was shown, the ratio of opened csv is 0% because no files were opened successfully. For all the csv the same error is raised: "<class 'urllib.error.URLError'>". Indeed, if any URL of the data

file is paste in the search engine, the return page is courtesy page that says: "This site can't be reached - dati.lazio.it took too long to respond". This error, as we can see in 4.5, occurred 43 times in all the data sets and it is the most common.

Other critical data set is the one that contains the csv of the Minstry of Health where 40% of the data files were zip file (even if they were labeled as csv). So, those files are downloaded but the algorithm can't process them because they are not single csv files. This error is the second most common one with 12 occurrences but all those belongs to the data set of Ministry of Health. So, even if this error has occurred several times, it is not popular among the tested data set.

The last 2 data set that have a low ratio are the ones referred to: region Emilia Romagna and region Basilicata. About the first one, all the opening errors are "<class 'urllib.error.URLError'>". So, as for the data set of the region Lazio, some URLs can't be opened. Instead, about the region Basilicata, those errors were raised:

- 1 error "<class 'urllib.error.URLError'>"
- 2 errors "HTTP Error 404: Not Found"
- 8 errors: "HTTP Error 403: Forbidden". As discussed in 3.4.4, this error was partly resolved altering the user-agent. Even so, for some data files of the region Basilicata the error is raised anyway. We have decided not to investigate further more.

4.3.2 Measure: COM-I-1 DevA

This QM refers to the ratio of non-null values over the data files. As it can be shown in figure 4.7, the data are sparser than the previous measure and the value is always greater than 0; so there were always some or several null values. Generally, around 97% of data is not null value and it seems that the ministries as MIUR, MIBACT, INPS have the best results for this metric.

The data set of Veneto region, instead, has an average of 22,9% of null data and it is the data set that performs worst for the metric COM-I-1 DevA. It is interesting, though, to see that from figure 4.3, the maximum value of that metric in the whole data set is just 0.5, while other public authorities have greater maximum value. Indeed, looking directly at the data files measure in figure 4.8, it is possible to see that in several csv this measure is quite high so it a problem that affect most of the data file of the Veneto region.

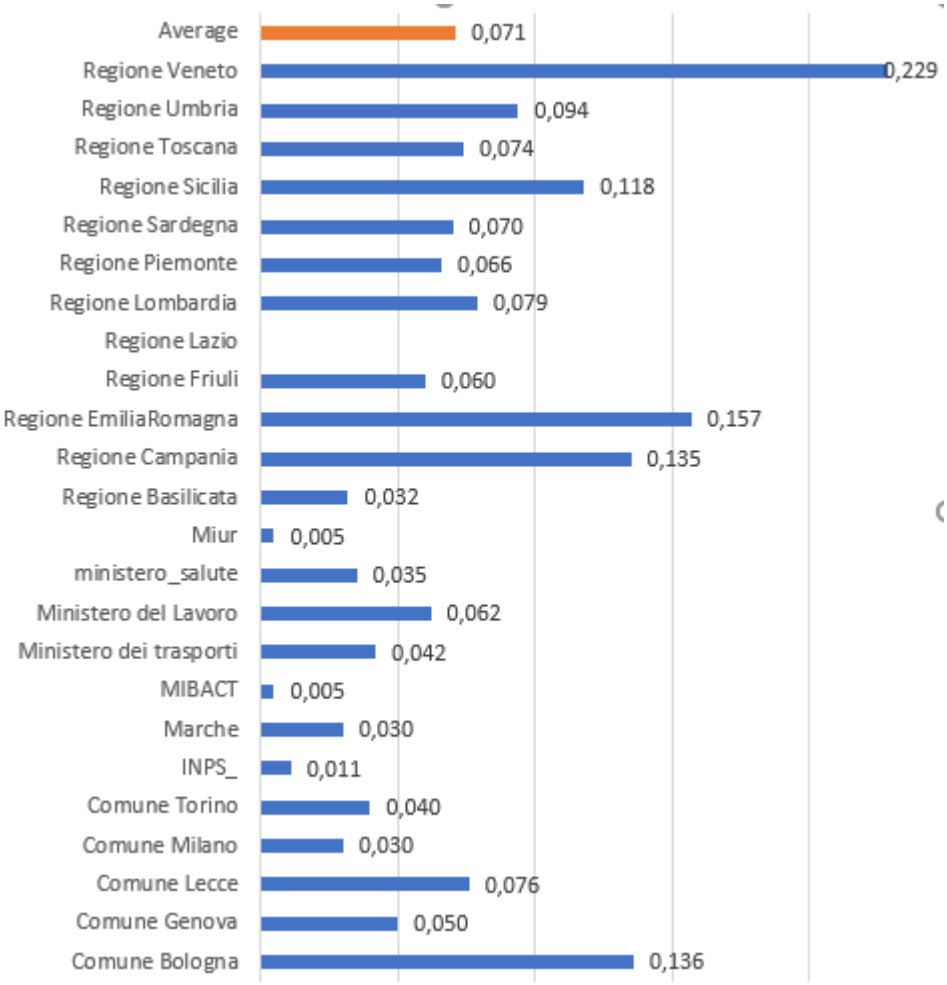

Figure 4.7. Visualization of QM Com-I-1 DevA

Instead, for the other critical cases as Region Emilia Romagna, Region Campania and municality of Bologna, in their data set the measure values are sparser and only in a few dataset the QM is high as shown in figure 4.9.

Finally, it interesting to notice that for the data set of region Emilia Romagna, in some data file the number of null values is high because entire rows or sub-rows are full of null values. A common line presents in several data files as [9] is the following: “;;;;;;” where “;” is the delimiter. This characteristics is also reflected in the measure Com-I-1.

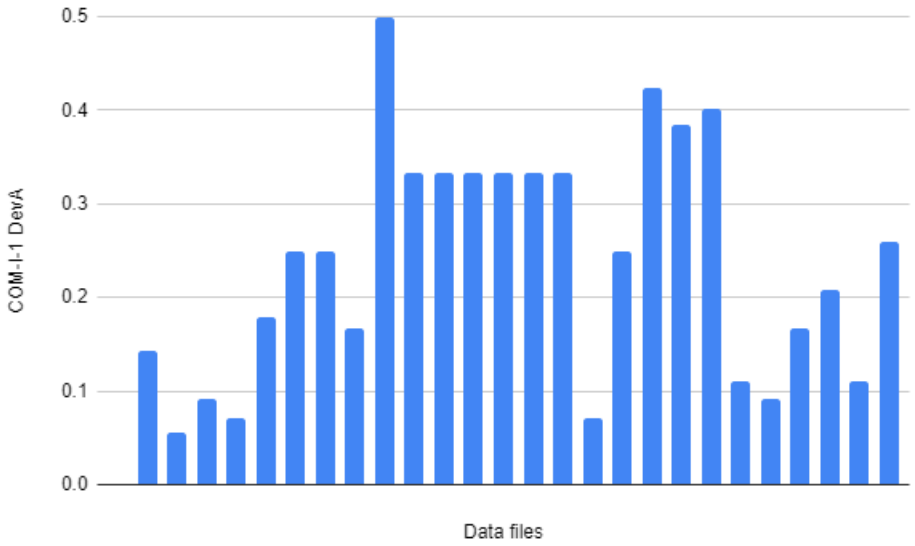

Figure 4.8. Visualization of the measure COM-I-1 in the data files of the region Veneto

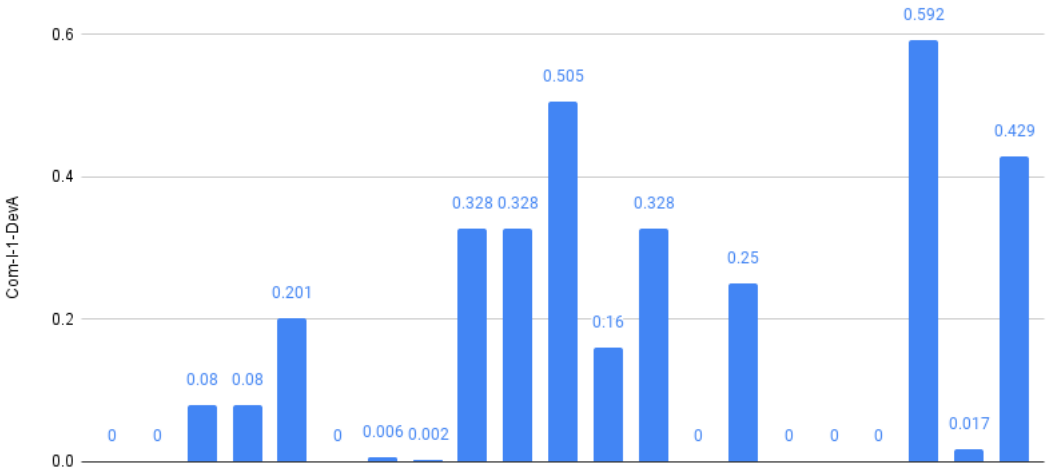

Figure 4.9. Visualization of the measure COM-I-1 in the data files of the region Emilia Romagna

4.3.3 Measure: Com-I-5

As it is shown in figure 4.10, all the data set have a good result in this metric that, as reminder, indicates the ratio of the rows that are not full of null.

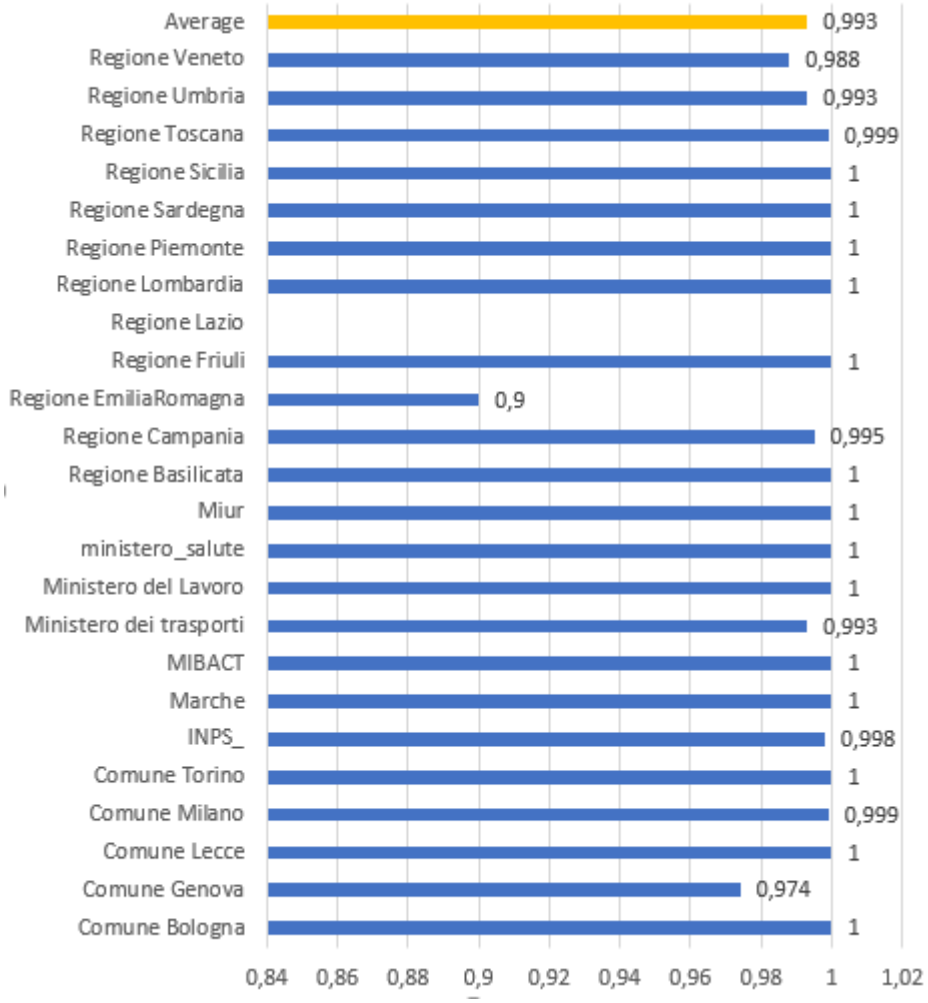

Figure 4.10. Visualization of QM Com-I-5

The only public authority that performs quite below the average is the region of Emilia Romagna. If we try to visualize the measure among all the data files, as in figure 4.11, we notice that this QM is critical only on 5 csv file, so it's not a general problem that affect the whole dataset.

It is also interesting to notice that the region Emilia Padana performs badly also for the measure Com-I-1 DevA. So, we can conclude that in the files of the region Emilia Romagna, the null values are not sparse over the

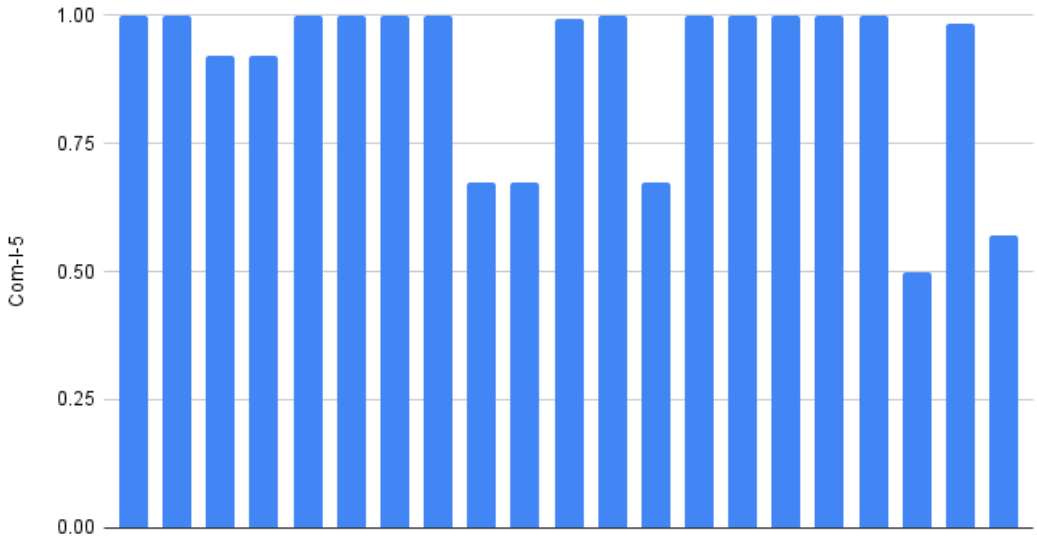

Figure 4.11. Visualization of the measure COM-I-5 in the data files of the region Emilia Romagna

csv but are presents in a few rows that are completely made of null values.

Finally, from figure 4.4, it is shown that the minimum value for this metric is referred to a data file [10] of the municipality of Genova. The measure suggests that near 75% of the rows of the data file are full of null values. Indeed, if the file is downloaded and open, 34 rows are ";;;" so they are made full of null; while only 12 rows have at least one value.

4.3.4 Measure: Acc-I-4

As shown in figure 4.12, all the data sets have a good result in this metric that, as reminder, measure the ratio of the number of outliers over the data files. MIBACT, Ministry of Transportation and municipality of Lecce are associated with the data sets with the highest values but are not considered as critical because the percentage of outliers doesn't reach the 10%.

Even if this measure has good results; from figure 4.3 the QM reaches especially high values for the municipality of Bologna and MIBACT data sets. Indeed, in the first case, in a data file [11] a numeric column is composed by several cells that all have the same number that is 100000. Due to the repetition of this number, the first quartile and the third quartile are all 100000, so $IQR = 0$. As result, all the numbers that are not 100000 are

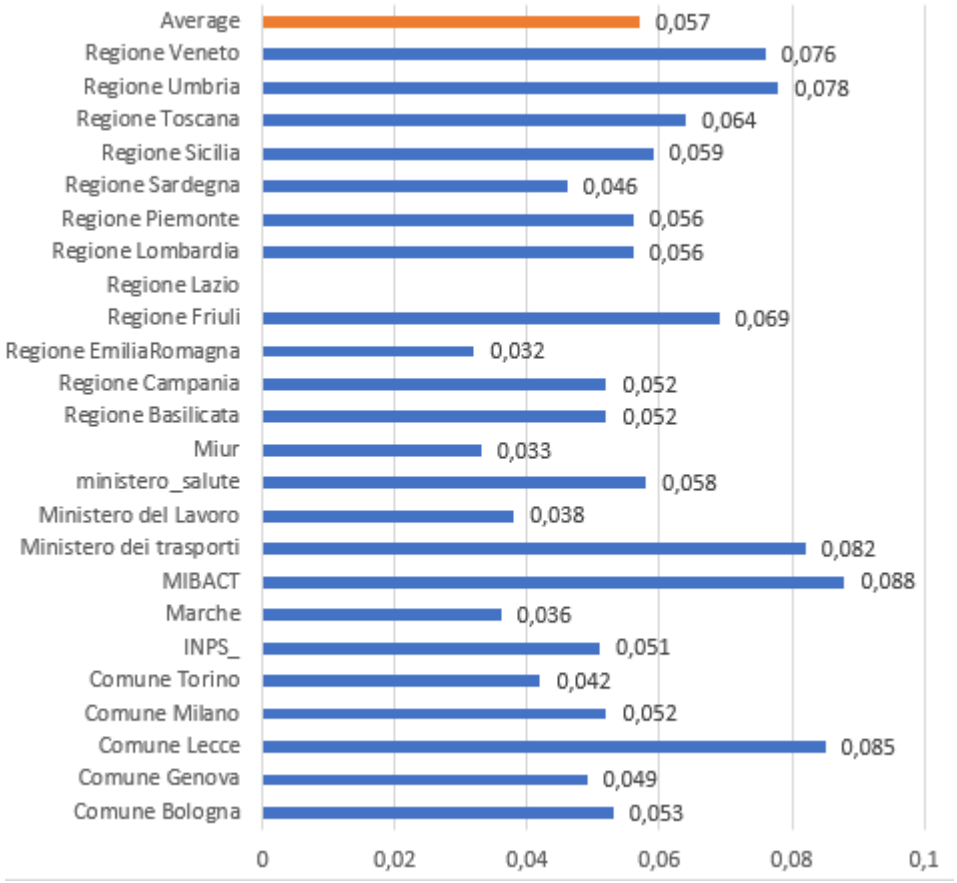

Figure 4.12. Visualization of QM Acc-I-4

considered as outliers by the algorithm as visualized in figure 4.13

The same scenario is present also for the data file [12] of the MIBACT authority. Indeed, in that file there is only a numeric column composed by a majority of 1, some 0 and other high values. Also in this case, the first and third quartiles are the same that is 1; so, all the numbers that are not 1 are considered outliers.

4.3.5 Measure: Con-I-3

This measure depends on the number of duplication values on the single attribute and by grouping 2 column together. As shown in figure 4.14, this measure is high for a limited number of authorities as Toscana region, municipality of Torino, municipality of Lecce, municipality of Bologna. In all the other cases, the value is below 2.

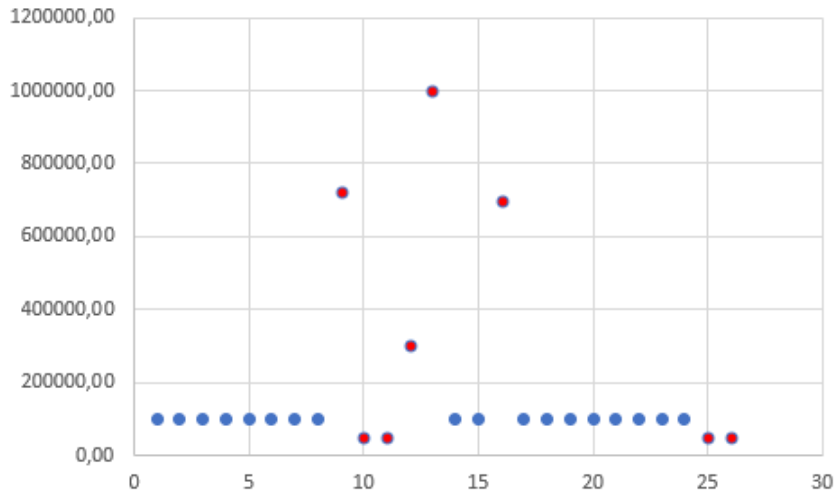

Figure 4.13. Visualization of numeric elements in the data file of the municipality of Bologna. The red point are the detected outliers.

Generally speaking, in these data set there are often a few data files where this measure is really high. A remarkable case is present in the data set of municipality of Torino where the metric of a data file [13] is 95. If the csv is open, we can see that there are several attributes that are boolean, so there are a huge number of repetition of 0 and 1. Indeed, for that data file, the number of duplication considering the single attributes is 7204, while the number of duplication grouping 2 attributes at time is 1046997. So, generally speaking, the QM is high for a data set because it is composed by several boolean attributes.

This rule is not true for a data file [14] of the region of Tuscany that, alone, change drastically the mean of the data set. Indeed, the Con-I-3 of the csv is 475.

This measure is really high because the data file has more than 100 columns and there is a sort of a pattern in the record. For example the sub row “94152640481, CONSORZIO LAMMA, 2015” is present in several lines. As a result, the number of number of duplication considering one attribute at the time is 5415, while the number of duplication grouping 2 attributes at the time is 20910873, roughly 21 millions.

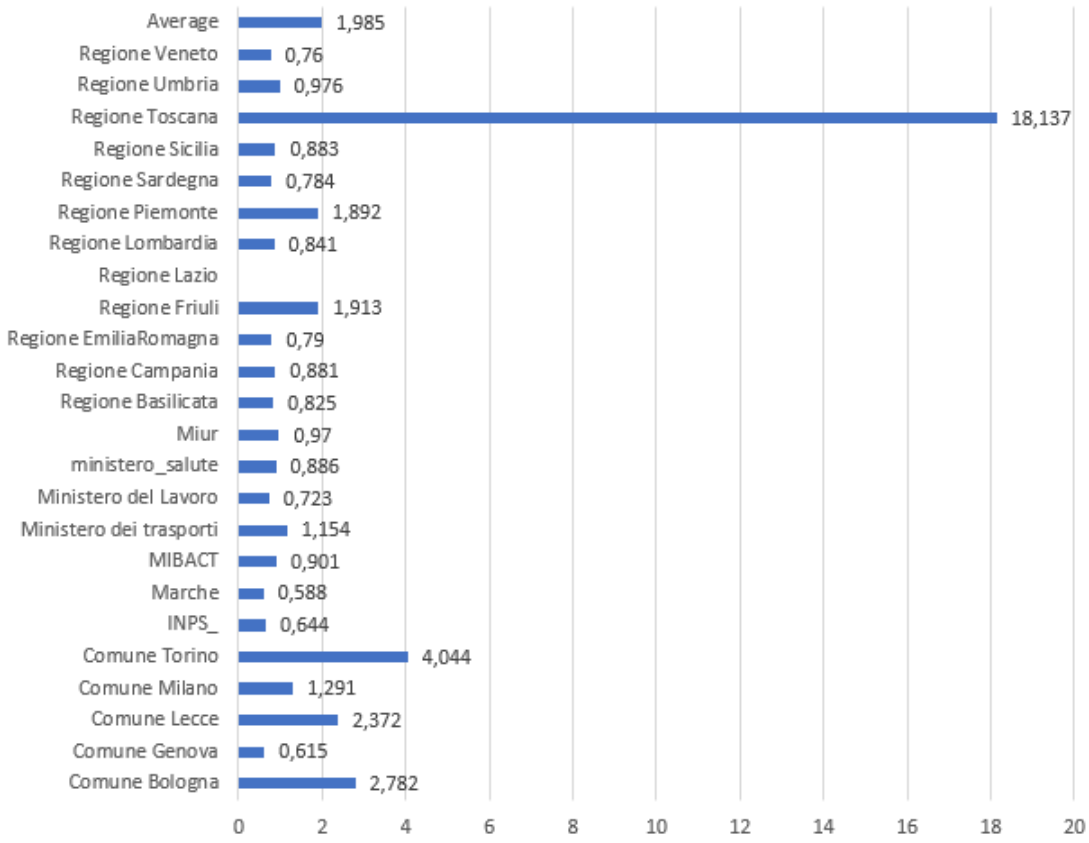

Figure 4.14. Visualization of QM Con-i-3

4.3.6 Measure: Con-I-2 DevB

Also for this QM, the results are over all good as shown in figure 4.15. There is only one critical case associated with the Ministry of Work where, in average, near the 3% of the data are inconsistent with the data type of the column where they are stored.

If the single data files of that data set are analyzed, there are just some CSVs where the Con-I-2 value is higher than the average but all the values are below the 10% as shown below in figure 4.16

Instead, from the image 4.3, we can see that this measure has reached an high value in a data file of the municipality of Lecce. Indeed, in that file [15], it can be spotted a strange behaviour: the null value is indicated with the symbol "-". The majority of the columns are numeric and that null value, that is considered as a string, is often present. As result, all those null values are counted as type inconsistency, so the measure is really high.

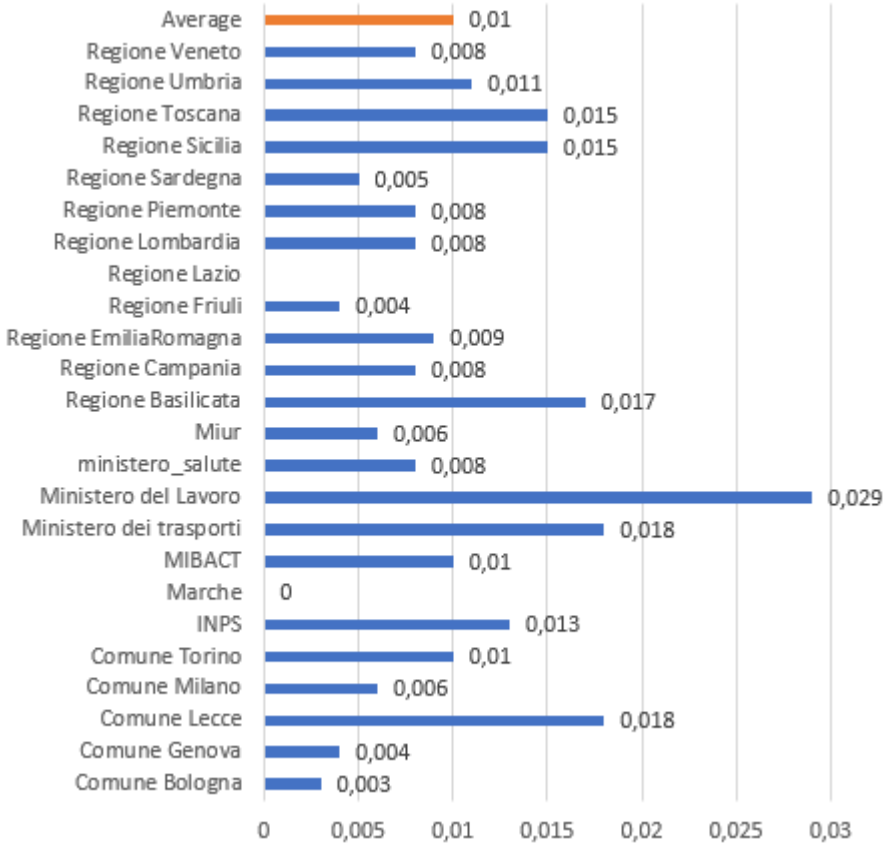

Figure 4.15. Visualization of QM Con-i-2 DevB

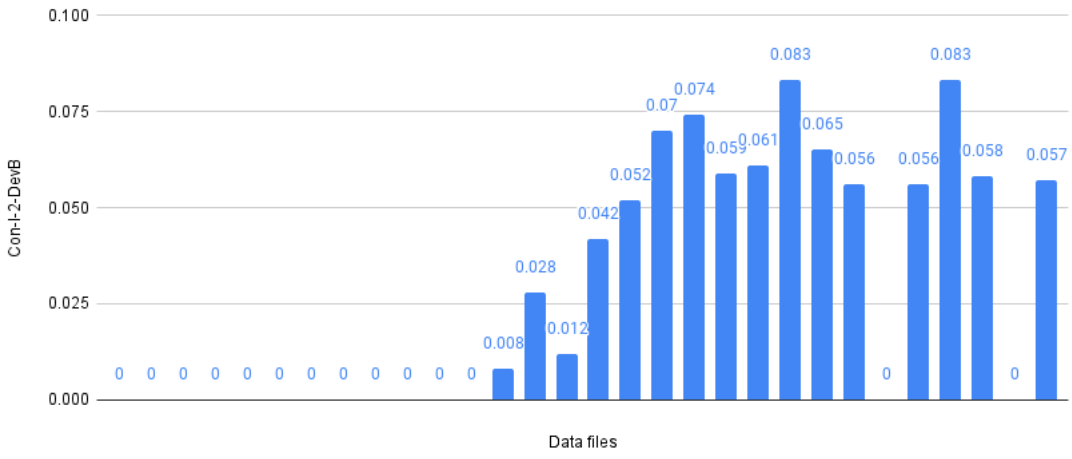

Figure 4.16. Visualization of the measure CON-i-2 in the data files of the Ministry of Work

4.3.7 Measure: Con-I-4 DevC

This QM measures the consistency of the csv structure over the file. As it can be seen from figure 4.17 the results are generally good but the measure about region Sicilia is really low.

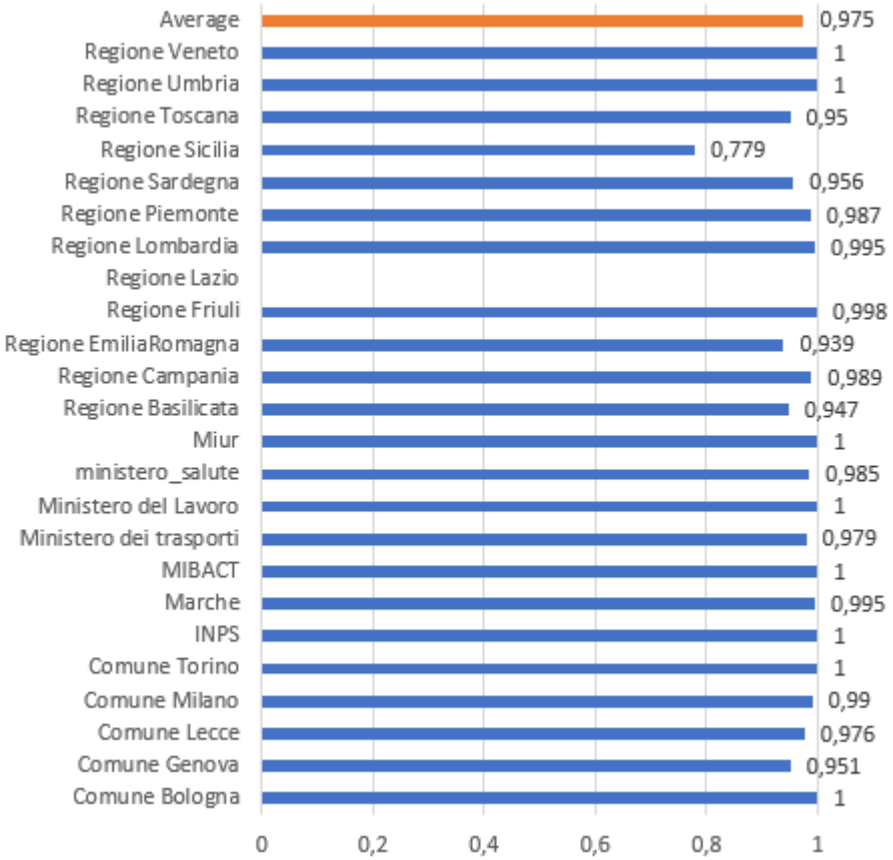

Figure 4.17. Visualization of QM Con-i-4 DevC

Analyzing the data files of the region Sicilia in figure 4.18, we can spot that there are some CSVs where the measure Con-I-4 is extremely low, near 0. Some of these cases are now discussed:

- Data file 5 [16]: the QM for this file is 0. Indeed, all the rows have a number of cells that is below the number of attributes in the header. This happens because, from an hint, all the information about one element that should be in one row is, instead, divided in 4 lines. For this reason, each line presents only some data and the standard CSV architecture is not present in this file.

- Data file 14 [17]: also in this case, from the row 5, all the lines have fewer cells. So, there are not null values; entire cells are not presents so the architecture consistency is really low.
- Data file 13 [18]: as for the first case, also in this file the information that should be present in just one line is, instead, divided in several rows. As result, several lines have a number of cells that is fewer than the number expected.

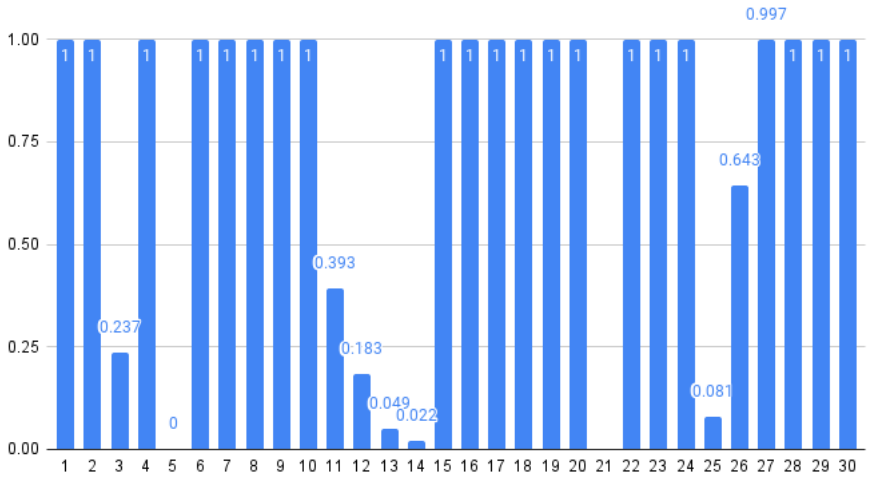

Figure 4.18. Visualization of the measure CON-i-4 DevC in the data files of the Region Sicilia

The same scenario seen in the data file 5 and 13 of the Region Sicilia, is also present in a file [19] of the region Sardegna where the QM is 0,011.

Instead, the opposite scenario is present in the data file [14] of the region Toscana where from the 18-th row, each line presents a greater number of cells than the number expected. It is also curious to notice that this file is the same analyzed in the section about measure Con-I-3.

4.4 Final considerations

4.4.1 Can Open

This is, maybe, the most important metric because if the data file can't be automatically opened by a software, it is impossible to make any analysis. As pointed out in figure 4.6, roughly 1 file every 10 can't be opened. This

is unacceptable because open government data should be all accessible for transparency and analysis purpose by experts or research centers.

As discussed previously, in more than half of the cases, the problem was in the URL of the data file. Indeed, some URLs don't return any page due to timeout or because the page does not exist or due to service unavailability. Every public authority should check periodically the availability of their public data files and, in case of error, remove it and create a new URL.

Another cause of error was due to the data file format. Indeed, the algorithm accept CSV file, so only the files labeled as CSV were taken in consideration. Despite this, in same cases the file format was not csv, beside it was zip or the file hasn't any format at all. Also for these cases, every authorities should check the integrity of the data file format and labeled the file correctly.

4.4.2 Con-I-1 DevA

This metric is about null values and is not in the scope of this work determine if the absence of data is relevant or not for the understandably and usability of the single data file. So, even if in some data set the number of null values is high, it can't be considered critical neither tolerable.

The only consideration that can be made is the following: generally, the number of outlier for this measure is really low. So, quite all the data files lay in the same range for this measure. Indeed, the null value issue is globally present in any data file and, during the tests, the number of data file that had no null value was low.

4.4.3 Com-I-5

In this case, as we have seen previously, the performs are generally high for the data set and there are a few exceptions. This is also true at the single data file level where quite all the CSVs have 1 but there are a few files with a low measure that, as result, reduce the Com-I-5 average for the public authorities associated with these data file (as region Emilia Romagna)

4.4.4 Acc-I-4

Also for this measure, the overall value is good but any consideration can't be done. Indeed, the algorithm detect the potential outliers but can't decide whether that value is an error or not. The number of data files with an high

value for this measure is small and these files should be checked manually by a domain expert in order to decide if the detected outliers are errors so they should be, eventually, deleted.

4.4.5 Con-I-3

As it was shown in the previous section, this metric can vary a lot on different data set. Generally the value is low because the majority of the OGD are composed by a limited number of column and rows. However, in some CSVs the number of columns is high (>10) and also the number of rows (>100000); as result this metric can reach really high value as 475. Also, the value can increase a lot if there are 1 ore more boolean attributes.

4.4.6 Con-I-2 DevB

The results about this measure are overall very good; indeed on average only 1% of the data has inconsistency in the type of its column. This QM is a critical one because if a value has type inconsistency, it can lead to error or unexpected results when its attribute or when the entire data file is analyzed to generate insights. In those cases, a post processing is required in order to clean out all these inconsistent data that are potential errors. As we have seen in the first chapter, the cost of cleaning data is 10 to 100 times bigger than the cost of preventing errors. For this reason, the public authorities should assess this measure before submitting a new data file and adjustments should be made if the QM is high.

4.4.7 Con-I-4 DevC

Also this measure is critical. If a file has an high value that is not 1, it just means that some data could be lost (if the number of cells is greater in some lines) or will be added null values (if the number of cell is smaller).

Instead, if the value is below 0.5%, it probably means that there is a problem at the architectural level as it was shown in some tests. In these cases, the file is not reliable so it should not be used for analysis purposes or to generate insight. Also this metric should be assess by the public authorities in order to detect irregularities in the file structure and, eventually, resolve them.

Chapter 5

Conclusions

5.1 Conclusions

Data quality is a broad notion with many interpretations, but its relevance is clear for both industry and government. In this thesis we have firstly defined the concept of data quality. why it is important nowadays and the challenges associated with this topic. Among these, it was shown that currently is not present a generalised algorithm that can assess the data quality independently of the data file. So, the aim of the thesis was this one: design and develop an algorithm that could assess automatically the data quality of any csv file.

In chapter 2, we have discussed about ISO 25012 and 25024 that were used as standard in order to define measurements through witch assessing the data quality. Those measures are the COM-I-5, ACC-I-4 and CON-I-3. Other 3 measure were altered from the original ones so that they could be applicable in the algorithm: COM-I-1 DevA, CON-I-2 DevB, CON-I-4 DevC.

In chapter 3, the algorithm is presented and discussed. In details, the core idea is exhaustively presented, the input and output of the algorithm are explained and it is technically described the implementation of every data quality measure. Lastly, some challenges faced during the implementation were highlighted in order to explain some technical decisions that were taken during the fine tuning of the algorithm.

In the thesis there was also a practical application of the algorithm that is used to asses the data quality of the Italian open government data sets. Indeed, 25 data files were randomly taken from several public authorities as region, municipality and ministries. The obtained results are described and they are also analyzed in order to highlight some anomalies on the data

set and try to find the causes behind them. Indeed, some considerations and suggestions are reported in order to avoid, in the future, the same data quality issues found in the files.

5.2 Future implementations

As said many times in this work, the main limitation of this algorithm is that it can only work with CSV file. As continuation of this work, the algorithm should become even more general and should handle other data format as JSON and XML.

Indeed, we also have to be aware that in the future the majority of the data files will be unstructured, so of format JSON or XML. Of course, the absence of a structure is a great complication for both data analysis and data quality assessment but in the actual state of art there are some libraries or technology that can help in handling those type of data sets.

Bibliography

[1] The Economics, *The world's most valuable resource is no longer oil, but data*, https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data (accessed October 25 2021)

[2] IBM Big Data & AnalyticsHub, *The Four V's of Big Data*, https://www.ibmbigdatahub.com/infographic/four-vs-big-data (accessed October 25 2021)

[3] Yannick Saillet, *How to quantify Data Quality?*, https://towardsdatascience.com/how-to-quantify-data-quality-743721bdba03 (accessed November 3 2021)

[4] Richard Farnworth, *The Six Dimensions of Data Quality — and how to deal with them*, https://towardsdatascience.com/the-six-dimensions-of-data-quality-and-how-to-deal-with-them-bdcf9a3dba71 (accessed November 3 2021)

[5] Susanne Morris, *Data Storage: JSON vs. CSV*, https://coresignal.com/blog/json-vs-csv/ (accessed November 7 2021)

[6] Md Sohel Mahmood, *Practical implementation of outlier detection in python*, https://towardsdatascience.com/practical-implementation-of-outlier-detection-in-python-90680453b3ce (accessed November 7 2021)

[7] Yuvraj Wadhwani, *What Is a 403 Forbidden Error (and How Can I Fix It)?*, https://www.howtogeek.com/357785/what-is-a-403-forbidden-error-and-how-can-i-fix-it/#autotoc_anchor_3 (accessed November 10 2021)

[8] Municipality of Napoli, http://dati.cittametropolitana.na.it/dataset/0f60e074-7004-41b5-a738-530fbd7c39a7/resource/3673af67-3399-46ba-b648-87b581763d83/download/entrate-2020.csv

[9] Region Emilia Romagna, http://www.comune.fe.it/3566/attach/opendata/docs/residenti_per_titolo_di_studio_2001_2017.csv

[10] Municipality of Genova, http://dati.comune.genova.it/sites/default/files/manif_fiera_2015_0.csv

[11] Municipality of Bologna, https://opendata.comune.bologna.it/api/v2/catalog/datasets/scuole-di-quartiere-interventi/exports/csv
[12] MIBACT, http://dati.san.beniculturali.it/dataset/csv/conservatore-ente_di_istruzione.csv
[13] Municipality of Torino, http://www.comune.torino.it/opendata/turismo/reg_ostelli_2017.csv
[14] Region Toscana, http://www.lamma.rete.toscana.it/sites/all/files/doc/consorzio/trasparenza/2015_procedimenti_amministrazione_bis.csv
[15] Municipality of Lecce, https://docs.google.com/spreadsheets/d/e/2PACX-1vTgV_NUVs8A_KtiLApnJkbANymwo1w_0vQxfA9fj7KrdUayCE3hJoMwj8DyUED-ah9OR5I374O7Vhc-/pub?gid=1543039375&single=trueoutput=csv
[16] Region Sicilia, https://dati.regione.sicilia.it/dataset/50bd3916-4b81-4573-a116-b4695ba73027/resource/facea51e-121d-4134-b99a-aae99812a69a/download/elencoopereincompiute2019rif2018.csv
[17] Region Sicilia, https://dati.regione.sicilia.it/dataset/32c11a40-56f3-459c-b00d-69a1aa5cffd0/resource/24555a79-d3e6-4f0b-9bfa-7788c8c6ddb9/download/58.csv
[18] Region Sicilia, http://93.41.160.42/BdProcedimenti/consulta/tmp/Procedimenti.csv
[19] Region Sardegna, http://dati.regione.sardegna.it/dataset/0529e1a3-4e0a-4bb3-9dd3-91adf43704f1/resource/b8e296eb-8674-4ed1-b5fb-0bdfeeea37b4/download/ufficicorpoforestale.csv