Politecnico di Torino

Master's Degree Course in Computer Engineering
Artificial Intelligence and Data Analytics

Master's Degree Thesis

A massive analysis of Italian open data quality

Supervisors
Prof. Antonio Vetrò
Prof. Marco Torchiano

Candidate
Kristi Gjerko

Academic Year 2023-2024

Abstract

The fast-paced diffusion of Open Government Data in the last few years in Italy has highlighted the need for a high standard of data quality to ensure that this data can be used to effectively support decision-making, transparency, and innovation. This thesis focuses on the design and implementation of a tool that allows for a massive analysis of the quality of Italian open data as published on the dati.gov.it portal. The objective is not to simply perform a one-time analysis but to provide a system that streamlines the entire process from data collection to quality evaluation. The primary goals are to automate the discovery, normalization, and quality assessment of a large number of datasets minimizing the manual work of the users and providing them with an interface that facilitates the analysis of the results.

One of the research areas is that of open data, a pillar of modern transparency and innovation initiatives. Open data is defined as data that can be freely used, modified, and shared by anyone for any purpose. Open Government Data is a type of open data that is collected, produced or paid for by the public sector. In Italy, the dati.gov.it portal serves as the main repository for OGD, hosting thousands of datasets across various domains. This data has the potential for improving governance, economic growth, innovation and public trust. However, its utility is closely tied to its quality, which leads to the other main research area of this thesis: data quality.

Data quality is a key component of the usefulness of information derived from data in a time when the quantity of data handled by computer systems is increasing worldwide and most business processes depend on it. The ISO/IEC 25012 and ISO/IEC 25024 standards are used as a basis for the study and implementation of the employed data quality model and measures. Data quality is not only a technical necessity but also a strategic one. High-quality data leads to reliable and actionable decisions while poor-quality data leads to errors and inefficiencies. Recognizing the pivotal role of data quality in the world of open data is the first step in maximizing its value and impact.

The primary contributions of this thesis in the development of the tool include:

1. Automated process pipeline

   - Implementation: A scalable solution was created to automate the processes of data discovery, normalization, and quality assessment. Its two main drivers are the metadata on the dati.gov.it portal and the ISO/IEC 25012 and ISO/IEC 25024 standards. The examined data quality characteristics are accuracy, availability, completeness, consistency, and understandability. In its entirety, the tool integrates a collection of Python scripts, a PostgreSQL database, a Node.js server, and a React-based user interface.
   - Advantages:
     - Eliminates the need for manual effort in processing open data.
     - Supports continuous monitoring and evaluation of data quality.
     - Simplifies data analysis through interactive user interfaces.

2. Scalable and modular architecture

   - Implementation: The architecture employs Docker containerization for deploying components independently. Each service, including database, processes, server, and client, is encapsulated ensuring smooth operation and integration.
   - Advantages:
     - High scalability, allowing easy integration of new datasets and quality measures.
     - High robustness, thanks to effective logging and error handling.

3. Advanced data quality analysis

   - Implementation: The tool was applied to a large collection from the open data portal, enabling in-depth analysis of over 20,000 files.
   - Advantages:
     - Highlights critical quality characteristics for intervention.
     - Provides actionable insights for improving data quality practices.
     - Facilitates compliance with open data principles.
     - Promotes the reusability and reliability of open data.

This thesis brings attention to the potential of utilizing open data and the importance of data quality. For data to be used in decision-making processes or analyzed to gain some insight, it has to maintain a high quality standard.

Analyzing the data quality results through the tool reveals some strong areas and some areas that need improvement, reinforcing the importance of continuous monitoring and refinement. The high availability and understandability scores show good levels of accessibility and clarity for the open data which is necessary for this data to be transparent and usable. The completeness and consistency also show encouraging scores reflecting good practices in the preparation and maintenance of the data. However, some accuracy and consistency measures suggest that there is still room for improvement.

The measurements can be extended to new quality characteristics if a complete and more detailed metadata catalog were to be available. Having more data about the data and its life-cycle can pave the way for the development of more quality measures and a deeper level of analysis. By providing a scalable, automated, and user-focused solution, this tool makes a significant contribution to advancing open data practices, ensuring that the promise of transparency and innovation is matched by the quality of the data itself.

Acknowledgements

I would like to thank Prof. Antonio Vetrò, Prof. Marco Torchiano, and Dr. Marco Rondina for their guidance and support in completing this thesis.

I would like to thank all the professors, staff, and colleagues at Politecnico di Torino for their dedication and companionship that have shaped me as an engineer over the last six years.

I would like to thank my colleagues at Irion for their mentorship and encouragement that have shaped me as a professional over the last three years.

I would like to thank my family, relatives, and friends for their love and care that have shaped me as a person throughout my life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and background

1.1 Open Data

According to the Open Definition [1]:

> Open data and content can be freely used, modified, and shared by anyone for any purpose

Open (Government) Data [2] refers to the information collected, produced or paid for by the public bodies (also referred to as Public Sector Information) and made freely available for re-use for any purpose. The Directive on the re-use of public sector information provides a common legal framework for a European market for government-held data. It is built around the key pillars of the internal market: free flow of data, transparency and fair competition. It is important to note that not all public sector information is Open Data.

The benefits of Open Data are diverse and range from improved efficiency of public administrations, economic growth in the private sector to wider social welfare. The economy can benefit from an easier access to information, content and knowledge, in turn contributing to the development of innovative services and the creation of new business models. For the 2016-2020 period, the cumulative direct market size of Open Data was estimated at 325 bn EUR. The higher demand for personnel with the skills to work with data lead to an estimated number of 100,000 Open Data jobs in 2020. Greater efficiency in processes and delivery of public services can be achieved thanks to cross-sector sharing of data, providing faster access to information. The cost savings for the EU28+ in 2020 were estimated to equal 1.7 bn EUR.

1.1.1 Italian OGD

dati.gov.it [4] is the Italian national catalog of open data and metadata of the public administration. This portal serves as the principal tool for the research and access of the data published according to the open data paradigm, in compliance with the provisions of Art.9 of Legislative Decree no.36/2006 (on the implementation of the European Directive on the reuse of public sector information). The project started in 2011 with the support of the Italian Government and it's been under the management of the Agency for Digital Italy since 2015. Open data represents an important pillar of open government and innovation so many initiatives have been launched for the technological evolution of the data catalog. The data.gov.it portal also contributes to the European portal data.europa.eu.

The publishing and updating of the data is the result of a process coordinated by the Agency for Digital Italy and carried out in collaboration with all the public entities that produce open data. The portal provides a series of instruments that have been used in this thesis and that allow exploring the topic of open data, improving their quality and, ultimately, encouraging their reuse.

1.2 Data Quality

While open data provides opportunities for transparency and innovation, its impact depends heavily on the quality of the data. Data quality is a key component of the usefulness of information derived from data in a time when the quantity of data handled by computer systems is increasing worldwide and most business processes depend on it.

A common prerequisite to all information technology projects is the quality of the data which are exchanged, processed and used between the computer systems and users and among computer systems themselves.

Managing and enhancing the quality of data is important because of:

- the acquisition of data from organizations of which the quality of data production process is unknown or weak;
- the existence of defective data contributing to unsatisfactory information, unusable results and dissatisfied customers;
- the dispersion of such data among various owners and users. Data captured in accordance with the workflow needs of a single organization

often lack a coherent and integrated vision which is necessary to ensure interoperability and co-operation;

- the need for processing data which are not immediately re-usable because of semantic ambiguity or lack of consistency between such data and other existing co-related data;

- the co-existence of legacy architecture and computer systems with distributed systems designed and realized at different times and with different standards;

- the existence of information systems (such as the world wide web) where data change frequently and integration is a special issue.

The data quality model defined in the ISO/IEC 25012 [3] International Standard aims to meet these needs, taking into account that the data life cycle is often longer than the software life cycle; it could be used, for example, to:

- define and evaluate data quality requirements in data production, acquisition and integration processes;

- identify data quality assurance criteria, also useful for re-engineering, assessment and improvement of data;

- evaluate the compliance of data with legislation and/or requirements.

The detection of errors or inefficiencies due to data gives rise to enhancement and corrective interventions concerning data and other components of the system in which data reside, for example:

- data (e.g. redesigning, parsing, cleansing, enriching, transforming, matching);

- software (e.g. modifying source programs to implement consistency controls);

- hardware (e.g. upgrading a computer system to improve response time);

- human business processes (e.g. user training to avoid errors in the data entry process; improvement of accounting processes that manage data).

Data quality is not only a technical necessity but also a strategic one, especially in the context of open data, where transparency and usability are the main focus. High-quality data leads to reliable and actionable insights while poor-quality data leads to errors and inefficiencies. Recognizing the pivotal role of data quality in the world of open data is the first step in maximizing its value and impact. The framework used for the assessment of data quality of the Italian open data in this thesis is the one laid out by the international standards ISO/IEC 25012 and ISO/IEC 25024.

1.2.1 ISO/IEC 25012

The ISO/IEC 25012 [3] standard defines a data quality model that categorizes quality attributes into fifteen characteristics considered by two points of view: inherent and system dependent.

Characteristics	**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

Table 1.1. Data quality model characteristics

Inherent data quality refers to the degree to which quality characteristics of data have the intrinsic potential to satisfy stated and implied needs when data is used under specified conditions. From the inherent point of view, data quality refers to data itself, in particular to data domain values and possible restrictions, relationships of data values and metadata.

System dependent data quality refers to the degree to which data quality is reached and preserved within a computer system when data is used under specified conditions. From this point of view data quality depends on the technological domain in which data are used; it is achieved by the capabilities of computer systems' components such as: hardware devices (e.g. to make data available or to obtain the required precision), computer system software (e.g. backup software to achieve recoverability), and other software (e.g. migration tools to achieve portability).

1.2.2 ISO/IEC 25024

The ISO/IEC 25024 [5] standard provides a basic set of data quality measures generated by a measurement function. The measurement function calculates a value with a range from 0,0 to 1,0 where values closer to 1,0 mean that the requirements for better quality are increasingly met. The analysis carried out in this thesis uses the following quality measures:

- **Accuracy** measures provide the degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use.

  Accuracy can be measured from the "Inherent" point of view only. Accuracy implies in some cases that the values agree with an identified source of validated information.

- **Completeness** measures provide the degree to which data associated with a target entity has expected values for all related properties of target entity in a specific context of use.

  Completeness can be measured from the "Inherent" point of view only. Completeness can be measured on a single attribute, or on the values of other attributes within a record or message.

- **Consistency** measures provide the degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use. They can be either or both among data regarding one target entity and across similar data for comparable target entities.

  Consistency can be measured from the "Inherent" point of view only.

- **Understandability** measures provide the degree to which data has attributes that enable it to be read and interpreted by users, and are expressed in appropriate languages, symbols and units in a specific context of use.

  Understandability is measured both from "Inherent" and "System dependent" point of view.

- **Availability** measures provide the degree to which data has attributes that enable it to be retrieved by authorized users and/or applications in a specific context of use.

  Availability can be measured from the "System dependent" point of view only.

Chapter 2

Architecture

The architecture of the tool is structured in five key components: the external data source, a pipeline of automated processes, a database for data storage, a server for API interaction, and a client for user visualization. This modular architecture and the containerization of its components through Docker provide good levels of performance, robustness, scalability, and maintainability.

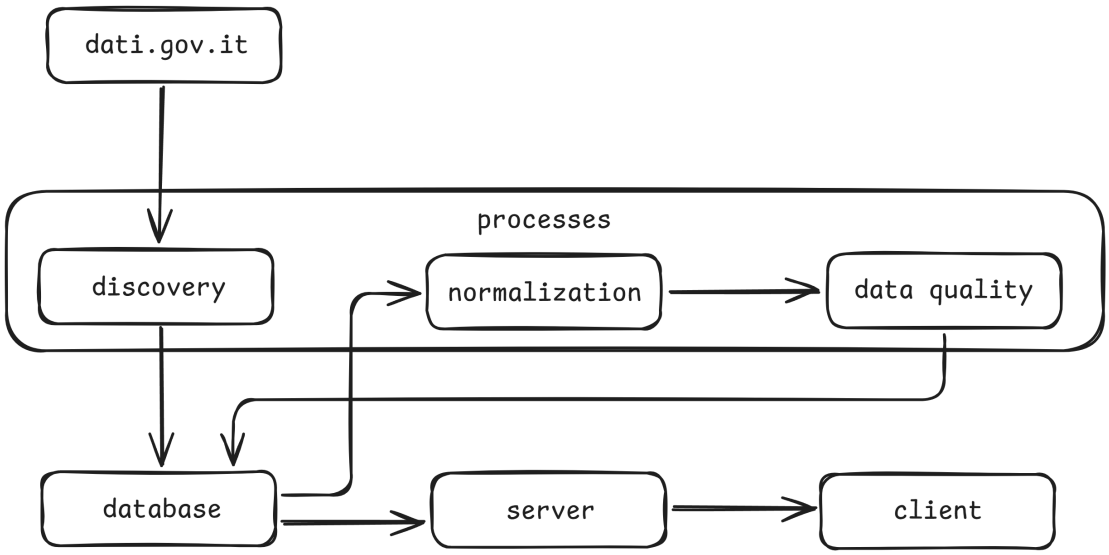

Figure 2.1. Tool architecture, components, and data flow

2.1 Data Source

The https://dati.gov.it portal makes its data available through an API interface and is structured as follows:

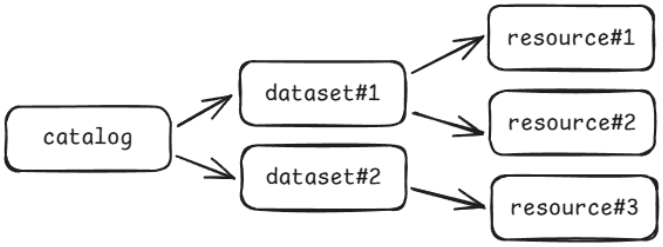

Figure 2.2. Italian open data portal structure

- The **catalog** contains a collection of datasets provided by various organizations. It acts as an index where users can find and access metadata and information about all available datasets.
- A **dataset** is an organized collection of data entries which includes metadata and references to the actual resources.
- A **resource** is a specific data entity within a dataset. Each dataset may contain multiple resources, even of different data formats such as CSV file, API endpoint, PDF document, etc.

The main endpoint to access the catalog is:

```
https://dati.gov.it/opendata/api/3/action/package_list
```

with the response:

```
{
  "help":    string,         //documentation
  "success": boolean,        //request status
  "result":  array[string]   //list of datasets
}
```

For each returned dataset it is possible to retrieve the complete metadata through the endpoint:

```
https://dati.gov.it/opendata/api/3/action/package_show?
id={dataset-id}
```

with the response:

```
{
  "help":    string,        //documentation
  "success": boolean,       //request status
  "result": {
    "id":           string,
    "name":         string,
    "organization": object,
    "groups":       array[object],
    "tags":         array[object],
    "resources": array[
      {
        "id":          string,
        "name":        string,
        "description": string,
        "url":         string,
      }
    ]
  }
}
```

For each returned resource it is possible to retrieve the corresponding file through the endpoint contained in the url field of the resources object in the response above. These files and the relative metadata will be the subject of data quality assessment.

2.2 Frameworks and Libraries

This section describes the utilized frameworks and libraries in terms of their main features and functions.

2.2.1 Database

A **PostgreSQL** database serves as the central repository that stores the processed metadata, data quality results, and process logs. PostgreSQL is a free and open-source RDBMS that is known for being reliable and having an extensive feature set. These characteristics make it ideal for handling the simultaneous read and write operations coming from the automated processes that continuously update the local metadata catalog and the execution of complex queries that transform and prepare the data for user interaction.

2.2.2 Processes

A collection of Python scripts handles the automated processes that are at the core of the tool.

1. The **data discovery process** utilizes the dati.gov.it APIs to identify and register potential datasets and resources by leveraging:
   - **urllib3**: A Python library for handling HTTP requests in a simple, efficient, and secure way. It provides customization for retries, timeouts, and error handling which are important aspects when dealing with multiple unknown sources.
2. The **data normalization process** takes the raw data as input and transforms it into structured formats that can be processed in Python. The main libraries it employs are:
   - **chardet**: A Python library used to determine the encoding of the raw data coming from the data discovery process. The encoding is crucial for reading the content of the raw data but it is initially unknown and varies across the different data sources.
   - **csv**: A Python library used to handle CSV files. Once the raw data has been decoded, this library helps determine the format of the CSV files by analyzing a sample and inferring the delimiter, quote character, header presence, etc.
   - **pandas**: A Python library used for data analysis and manipulation. After having determined the characteristics of the raw data in the previous steps, this library uses that information to transform the raw data into dataframes. A dataframe is a tabular data structure that can be manipulated easily and efficiently thanks to the extensive collection of functions provided by pandas.
3. The **data quality process** is the final step of this automated pipeline. It takes dataframes in input and calculates the data quality measures as per ISO/IEC standards by utilizing the following library:
   - **numpy**: A Python library used for numerical and scientific computing that integrates seamlessly with pandas and provides mathematical functions for working with arrays and matrices. It is fast and memory efficient due to its optimized C implementation. Good performance is very important considering the quantity of data that the tool has to process daily.

2.2.3 Server

The server is implemented to expose APIs for querying the stored metadata and data quality results. It serves as the bridge between the database and the client interface, ensuring secure and efficient data access. Here, the main adopted frameworks are:

- **Node.js**: An open-source and cross-platform runtime environment that allows to build server-side applications by using JavaScript. Its asynchronous and non-blocking I/O operations make it ideal for handling multiple database queries and API requests.
- **Express.js**: A lightweight web framework for the creation and configuration of API endpoints. Its features and built-in support for middleware setup provide an intuitive and developer-friendly alternative for building RESTful APIs.

2.2.4 Client

The client is built with the purpose of providing the user with an interactive and functional interface for data visualization and analysis. The key libraries used to implement it are:

- **React**: A JavaScript library used to build interactive user interfaces, particularly for single-page applications. Its support for reusable UI components and efficient rendering provide excellent performance for dynamic data-driven applications.
- **ECharts**: A rich chart library for creating interactive and visually appealing data visualizations such as histograms, line graphs, and box plots. The seamless integration with React simplifies the designing of charts in the application.
- **AG Grid**: A visualization library for displaying and managing tabular data. It provides dynamic and interactive data grids with sorting, filtering, and pagination that are easily integrated with React.

2.3 Docker Orchestration

Docker is a platform for the development, deployment, and management of applications through containerization. The code, dependencies, and runtime environment of the application are encapsulated in containers to ensure that it runs consistently across different systems. This eliminates compatibility issues and simplifies deployment, in particular for applications that rely on multiple components.

In the context of this tool, Docker was used to facilitate the orchestration and deployment of its core components: database, scripts, server, and client. Each component is encapsulated in a Docker container with the configurations defined in the corresponding Dockerfiles while their functioning as a whole is orchestrated via a docker-compose.yml file.

The database service is built from the official Postgres image and receives the database credentials as environment variables. A Docker volume is created to store the database files to ensure that data is retained even if the container restarts. Another volume stores the initialization script that is executed on container startup to create the database schema and tables. The internal and external ports where this service is exposed are also configured.

The script service is built with the necessary dependencies defined in the requirements.txt file. Once all the dependencies have been installed and the container is ready, it still has to wait before the automated processes can start. This is due to the fact that these processes perform read and write operations on the database so the wait-for-it.sh script ensures that the database service is up and running for the processes to be executed. It is possible to configure the processes' start times and durations through environment variables which does not require rebuilding the container.

The server and client services are more straightforward. Their internal and external ports are configured and their dependencies are installed as declared in the corresponding package.json files. The client receives the address of the server through environment variables while the server receives the address of the database. All the services are part of a shared Docker bridge network that enables the communication between them.

Chapter 3

Implementation

This chapter describes the technical implementation of the different components of the tool. It details how each of the components was developed and integrated and describes the overall approach for achieving the desired level of modularity, scalability, and automation. The source code is available in a dedicated public Git repository [6].

3.1 Database

The database is implemented using PostreSQL and stores all the necessary data to support the functioning of the tool. It is organized under the schema opendata and contains multiple interconnected tables that reflect the logical structure of the data model. The tables are:

1. **datasets**
   - **Purpose**: stores the metadata of the datasets
   - **Columns**:
     - **id**: (text, primary key) the identifier of the dataset
     - **name**: (text) the name of the dataset
     - **description**: (text) a textual description of the dataset
     - **organization**: (text) the organization that owns the dataset
     - **metadata_created**: (timestamp) timestamp indicating when the dataset metadata were created on the open data portal
     - **metadata_modified**: (timestamp) timestamp indicating when the dataset metadata were last modified on the open data portal

  - **dataset_modified**: (timestamp) timestamp indicating when the dataset was last modified
  - **row_registered**: (timestamp) internal timestamp indicating when the dataset metadata were first registered on the tool
  - **row_modified**: (timestamp) internal timestamp indicating when the dataset metadata were last modified on the tool

2. **resources**

- **Purpose**: stores the metadata of the individual resources
- **Columns**:
  - **id**: (text, primary key) the identifier of the resource
  - **name**: (text) the name of the resource
  - **description**: (text) a textual description of the resource
  - **format**: (text) the file format of the resource
  - **dataset**: (text) the identifier of the dataset containing the resource, references the column **id** of **datasets**
  - **url**: (text) the url where the resource can be accessed
  - **metadata_modified**: (timestamp) timestamp indicating when the resource metadata were last modified on the open data portal
  - **resource_created**: (timestamp) timestamp indicating when the resource was created
  - **resource_modified**: (timestamp) timestamp indicating when the resource was last modified
  - **row_registered**: (timestamp) internal timestamp indicating when the resource metadata were first registered on the tool
  - **row_modified**: (timestamp) internal timestamp indicating when the resource metadata were last modified on the tool
  - **row_processed**: (timestamp) internal timestamp indicating when the resource was last processed by the tool
  - **num_available**: (integer) the number of times the resource was available when attempted to be accessed by the tool
  - **num_accesses**: (text) the number of times the resource was attempted to be accessed by the tool

3. **measures**

   - **Purpose**: stores the metadata of the quality measures
   - **Columns**:
     - **measure**: (text) the identifier of the measure
     - **title**: (text) the name of the measure
     - **descr**: (text) a detailed textual description of the measure

4. **quality**

   - **Purpose**: stores the results of the data quality assessment process
   - **Columns**:
     - **id**: (text) the identifier of the resource processed for data quality, references the column **id** of **resources**
     - **acci3**: (numeric) the score of the data quality measure
     - **acci4**: (numeric) the score of the data quality measure
     - **avad1**: (numeric) the score of the data quality measure
     - **comi1**: (numeric) the score of the data quality measure
     - **comi5**: (numeric) the score of the data quality measure
     - **coni2**: (numeric) the score of the data quality measure
     - **coni3**: (numeric) the score of the data quality measure
     - **coni4**: (numeric) the score of the data quality measure
     - **coni5**: (numeric) the score of the data quality measure
     - **undi1**: (numeric) the score of the data quality measure
     - **tscreation**: (timestamp) internal timestamp indicating when the data quality scores for the resource were calculated
     - **execid**: (text) an identifier linking the scores to a specific execution of the process, references the column **id** of **log_exec**

5. **datasets_groups**

   - **Purpose**: associates datasets to their corresponding groups
   - **Columns**:
     - **dataset**: (text) the identifier of the dataset, references the column **id** of **datasets**
     - **grp**: (text) the group where the dataset belongs

6. **datasets_tags**
    - **Purpose**: associates datasets with descriptive tags
    - **Columns**:
        - **dataset**: (text) the identifier of the dataset, references the column **id** of **datasets**
        - **tag**: (text) the tag associated to the dataset
7. **log_exec**
    - **Purpose**: stores the execution logs for the processes of the tool
    - **Columns**:
        - **id**: (text) the identifier of the execution
        - **type**: (text) the type of process in execution
        - **msg**: (text) the step of the process and its result
        - **tscreation**: (text) internal timestamp indicating when the step was finished
8. **log_csv**
    - **Purpose**: stores the logs of the data discovery process for the CSV resources
    - **Columns**:
        - **id**: (text) the identifier of the resource, references the column **id** of **resources**
        - **msg**: (text) the outcome of the process, successful or error
        - **encoding**: (text) the encoding of the CSV file
        - **delimiter**: (text) the delimiter of the CSV file
        - **nr**: (integer) the number of data rows present in the file
        - **nc**: (integer) the number of data columns present in the file
        - **nl**: (integer) the number of text lines present in the file
        - **tscreation**: (text) internal timestamp indicating when the file was processed
        - **execid**: (text) the identifier of the execution where the file was processed, references the column **id** of **log_exec**

9. **log_quality**

   - **Purpose**: stores the error logs for the data quality process
   - **Columns**:
     - **id**: (text) the identifier of the resource processed for data quality, references the column **id** of **resources**
     - **msg**: (text) the error message
     - **tscreation**: (text) internal timestamp indicating when the error occurred
     - **execid**: (text) the identifier of the execution where the error occurred, references the column **id** of **log_exec**

3.2 Processes

The automated pipeline of processes is at the core of the tool. They are designed to handle the complex but repetitive aspects such as the collection, cleaning, and storing of the data. By leveraging the portal's metadata and a series of automated workflows, the tool is able to maintain an up-to-date database and a read-for-use interface so the user can immediately access the data quality results without having to handle anything. Delegating all the preparation operations to the processes means that the tool is reliable and scalable compared to depending on manual preparation. Having these steps handled by the tool in a totally automated way enhances its usability and efficiency and leaves the user only with the task of analysis of the provided insights.

3.2.1 Data Discovery

The data discovery process is the first step of the automated pipeline. It is responsible for identifying and registering datasets and resources from the open data portal. The process reads the metadata from the portal's API, compares it with the existing database records, and updates the database accordingly. A complete execution, going through all the available metadata, employs on average 4 hours and discovers circa 65.000 datasets and 210.000 resources.

There are three main scripts that make up the process: one for API interaction with the portal, one for database operations, and one for orchestrating the overall logic. The general workflow follows these steps:

1. retrieve from the API the list of all available datasets
2. retrieve from the API the detailed metadata and associated resources for each dataset
3. compare the retrieved metadata with the database records to determine the operations to perform
4. perform the operations on the database
5. log the execution progress, the number of processed datasets and resources, and eventual errors.

3.2.2 Data Normalization

The data normalization process uses the information collected by the data discovery process to retrieve the raw files and transform them into structured formats that can be analyzed. The process runs concurrently with the data quality process and it's able to normalize only the CSV resources. The general workflow follows these steps:

1. retrieve the file from the resource url and verify that it is a CSV
2. detect the encoding and decode the file content
3. extract a sample of lines from the file
4. use the csv.Sniffer to detect the CSV characteristics such as delimiter, quote character
5. clean the lines form leading or trailing whitespace
6. determine the CSV header
7. read the CSV content into a dataframe
8. log the execution progress, resource characteristics, and eventual errors.

3.2.3 Data Quality

The data quality process takes in input the normalized version of each resource and outputs the calculated data quality measure scores. The generalized algorithm it uses is an extension of a previous implementation [7]. Its workflow follows these steps:

1. retrieve the dataframe
2. compute the quality measures
3. update the database with the results
4. log the execution progress and eventual errors

The functions used for the evaluation group multiple measures together to be more efficient given the number of evaluations the process needs to perform daily. For particularly large resources, just going through them once is costly so the functions are organized in a way to minimize that as much as possible by calculating simultaneously all the needed metrics. The process goes through circa 1000 resources in an hour.

Every standard measure is assigned an identification code (ex: Acc-I-1) that consists of the following fields:

- **CCC:** abbreviated alphabetic code representing the quality characteristics (ex: Acc for Accuracy)
- **Z:** I for Inherent or D for System Dependent expressing the point of view of the data quality characteristic
- **Y:** serial number of sequential order within data quality characteristics and point of view

For the derived measures, the code (ex: Acc-I-1-IT-1) contains an additional two fields:

- **AA:** abbreviated aplhabetic code representing the country where the new derived measure is registered
- **v:** serial number of sequential order for measures with the same values for the previous four fields

The implemented measures are:

- **Acc-I-3**
  - **Name:** Data accuracy assurance
  - **Description:** Ratio of measurement coverage for accurate data
  - **Function:** X=A/B
    - A=number of data items measured for accuracy
    - B=number of data items for which measurement is required for accuracy
  - **Implementation:** This measure does not evaluate the quality of the data, but the thoroughness and application of the accuracy measures. It is a measure of the attention given to the accuracy matter. It can provide important context and put things in perspective given that there is only one other implemented measure for accuracy.

    Here, the data items are the cells of the dataframe. A is the number of cells that are used in the calculation of an accuracy measure. B is the total number of cells.
- **Acc-I-4-IT-1**
  - **Name:** Confidence of data set accuracy
  - **Description:** A low number of outliers in values indicates confidence in the accuracy of the data values in a data set
  - **Function:** X=1-A/B
    - A=number of data values that are outliers
    - B=number of data values to be considered in a data set
  - **Derivation:** This measure has been derived from the standard measure Acc-I-4 (Risk of data set inaccuracy) with formula X=A/B where for X, lower is better. The purpose of this derivation is to have a common semantic across all measures such that their values range from 0 to 1, where higher is better.
  - **Implementation:** An outlier is a value that is numerically distant from the rest of the values. Here, it has been calculated using the IQR method. It is possible to calculate outliers only for data values of numeric types.

    Here, the data values are the values contained in the cells of the dataframe. A is the number of values of numeric type that are

outliers. B is the total number of values of numeric type. The fact that this measure considers only numerical types highlights the need for the Acc-I-3 that provides the coverage for accuracy. In a data set made only of non-numerical types Acc-I-4-IT-1 would be 0 but this does not mean that the entire data set is inaccurate. In this case, Acc-I-3 would also be 0, helping understand that accuracy measurement was not possible.

- **Ava-D-2**
  - **Name:** Probability of data available
  - **Description:** The probability of successful requests trying to use data items during requested duration
  - **Function:** X=A/B
    - A=number of times that data items are available
    - B=number of times that data items are requested
  - **Implementation:** The availability of the data items is tracked in the moment when the data normalization process requests the data files to convert them to dataframes. The data items are deemed available if the corresponding endpoint responds to the request within a five second timeout.
- **Com-I-1-IT-1**
  - **Name:** Data set completeness
  - **Description:** Completeness of data items within a data set
  - **Function:** X=A/B
    - A=number of data items with associated value not null in a data set
    - B=number of data items of the record for which completeness can be measured
  - **Derivation:** This measure has been derived from the standard measure Com-I-1 (Record completeness). The purpose of this derivation is to have a common semantic across all measures such that their calculation is performed on the entire data set instead of single records.
  - **Implementation:** Here, the data items are the cells of the dataframe. A is the number of cells containing not null values. B is the total number of cells.

- **Com-I-5**
  - **Name:** Empty records in a file
  - **Description:** False completeness of records within a data file
  - **Function:** X=1-A/B
    - A=number of records where all data items are empty
    - B=number of records in a data file
  - **Implementation:** Here, the records are the rows of the dataframe. A is the number of rows with all cells containing not null values. B is the total number of rows.
- **Con-I-2-IT-1**
  - **Name:** Data type consistency
  - **Description:** Consistency of data type of the same attribute
  - **Function:** X=A/B
    - A=number of data items where the data type is consistent with the attribute
    - B=number of data items for which type consistency can be defined
  - **Derivation:** This measure has been derived from the standard measure Con-I-2 (Data format consistency). The purpose of this derivation is to have a more generalizable approach. The properties of the data format are not known so the data type which can be inferred from the file is used instead.
  - **Implementation:** Here, the data items are the cells of the dataframe, the attributes are the columns and the data type is the type of the column which can be string or numeric. The type of the column is inferred based on the most common type of its cells. Then the cells that are consistent and inconsistent with that type are counted.
- **Con-I-3-IT-1**
  - **Name:** Confidence of data set consistency
  - **Description:** Confidence of having consistency due to the absence of duplications of data values
  - **Function:** X=1-A/B

    - A=number of data items where exist duplications in value
    - B=number of data items considered
  - **Derivation:** This measure has been derived from the standard measure Con-I-3 (Risk of data set inconsistency) with formula X=A/B where for X, lower is better. The purpose of this derivation is to have a common semantic across all measures such that their values range from 0 to 1 and higher is better.
  - **Implementation:** Here, the data items are the cells of the dataframe. The measure calculates the duplications of single and pairs of cell values.

- **Con-I-4-IT-1**
  - **Name:** Data structure consistency
  - **Description:** Degree to which the data records are consistent with the data structure of the file
  - **Function:** X=A/B
    - A=number of data records that are consistent with the data structure
    - B=number of data records contained in the file
  - **Derivation:** This measure has been derived from the standard measure Con-I-4 (Architecture consistency). The purpose of this derivation is the absence of knowledge of the broader data architecture and data model. From the received file only a structure for its content can be inferred so the measure is modified to check the consistency of its records with the structure.
  - **Implementation:** Here, the data records are the rows of the file. The normalization process that infers the header of the file and its structure traces the number of total rows in the file and the number of total rows that are consistent with the data structure and can be normalized.

- **Con-I-5**
  - **Name:** Data values consistency coverage
  - **Description:** Coverage of consistency measurement of data values
  - **Function:** X=A/B

    - A=number of data items considered in consistency measurement of data values
    - B=number of data items for which consistency are measured
  - **Implementation:** This measure does not evaluate the quality of the data, but the thoroughness and application of the consistency measures. It is a measure of the attention given to the consistency matter. It can provide important context and put things in perspective given that of the three other consistency measures, only Con-I-3-IT-1 measures the consistency of the data values.

    Here, the data items are the cells of the dataframe. A is the number of cells whose values have been used in the calculation of a consistency measure. B is the total number of cells.

- **Und-I-1**
  - **Name:** Symbols understandability
  - **Description:** Degree to which comprehensible symbols are used
  - **Function:** X=1-A/B
    - A=number of data values represented by known symbols
    - B=number of data values for which symbols understandability is requested
  - **Implementation:** During the decoding phase the unknown symbols are replaced with the replacement character  . Here, the data values are the values contained in the cells of the dataframe. A is the number of values that contain the replacement character. B is the total number data values.

3.3 Server

The server acts as a means of communication between the database and the client. Its purpose is to retrieve and transform the data needed for an intuitive analysis through the user interface. It interacts with the database through SQL queries, allows requests coming only from the client, and handles errors by returning descriptive messages with HTTP status codes. All the endpoints are implemented as GET methods since the client is designed only for data visualization. As seen previously, the modification of the data is handled by the automated processes which interact directly with

the database. This separation of responsibilities helps maintain the integrity of the data while allowing for it to be accessed in a secure and reliable way.

The available API endpoints are:

1. **/api/home**

   This endpoint provides the overall statistics visualized in the homepage of the user interface. Its response includes:

   - the total number of discovered datasets and resources
   - the total number of CSV resources classified as valid, invalid, or unavailable
   - the trend of the statistics mentioned above for the last seven days
   - the average scores and descriptions of each data quality measure
   - the total scores of each data quality measure for the calculation of the distributions

2. **/api/group**

   This endpoint provides the total number and average score of each quality measure for the resources as aggregated by their associated groups.

3. **/api/tag**

   This endpoint provides the total number and average score of each quality measure for the resources as aggregated by their associated tags.

4. **/api/organization**

   This endpoint provides the total number and average score of each quality measure for the resources as aggregated by their publishing organization.

5. **/api/dataset**

   This endpoint provides the total number and average score of each quality measure for the resources as aggregated by their corresponding dataset.

6. **/api/resource**

   This endpoint provides the score of each quality measure for individual resources.

3.4 Client

The client consists in a user interface for visualizing and interacting with the metadata and the data quality results. The interface is implemented with React and provides a visually appealing and intuitive analysis experience for the user. It is composed of multiple views that use the server's APIs for receiving the data to then render the graphic components.

1. The **Toolbar** helps with the navigation between the different views. Each of the six buttons, with the characterizing icon and tooltip, corresponds to a view. The currently selected view is saved in a React state variable and its button is highlighted. An update is triggered on button click after which the view and the relevant components are re-rendered.

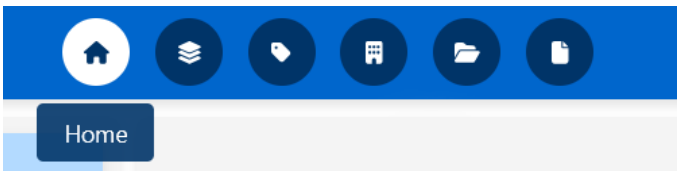

Figure 3.1. User interface toolbar

2. The **Homepage** is a dashboard that summarizes the overall status of the tool, from process performance metrics to data quality measures.

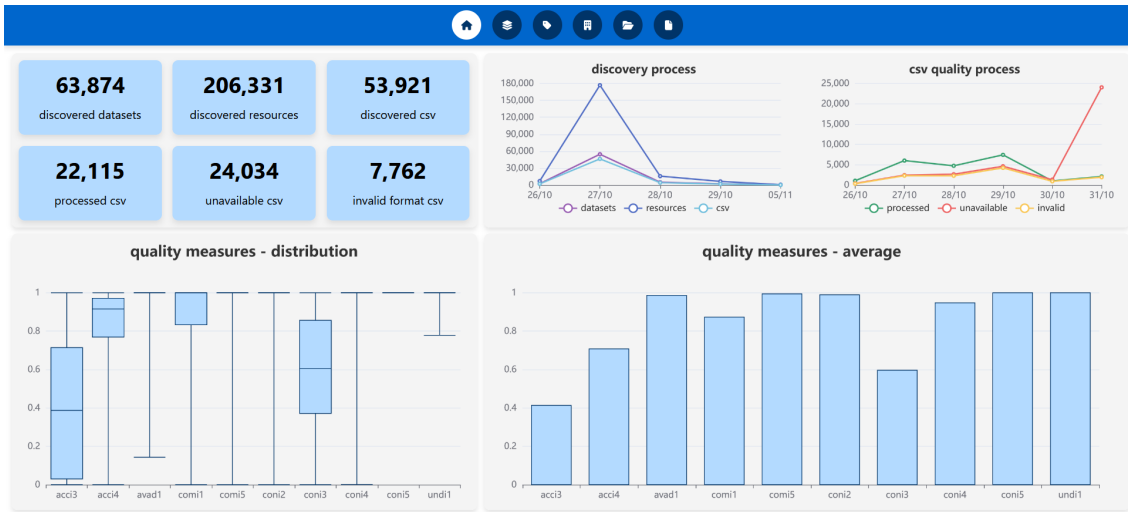

Figure 3.2. User interface homepage

The upper left section presents statistics on the datasets, resources, and CSV files. A card layout has been used for emphasis and readability.

The upper right section is made of two line charts that display the trend over the last 7 days for the discovery and quality process numbers.

The bottom section provides insight on the scores of the data quality measures. The boxplot on the left visualizes the distribution of the scores by highlighting min, max, median, and quartile values while the bar chart on the right visualizes the overall average scores.

The graphs are implemented from the ECharts library while the cards are custom made. The data needed used in the Home view is retrieved from the /api/home endpoint.

3. The **Dimension** views share a similar composition and functionality. The five dimensions are **Group**, **Tag**, **Dataset**, **Organization**, and **Resource**. Each view helps analyze the quality results from a different perspective based on the dimension the scores are aggregated by.

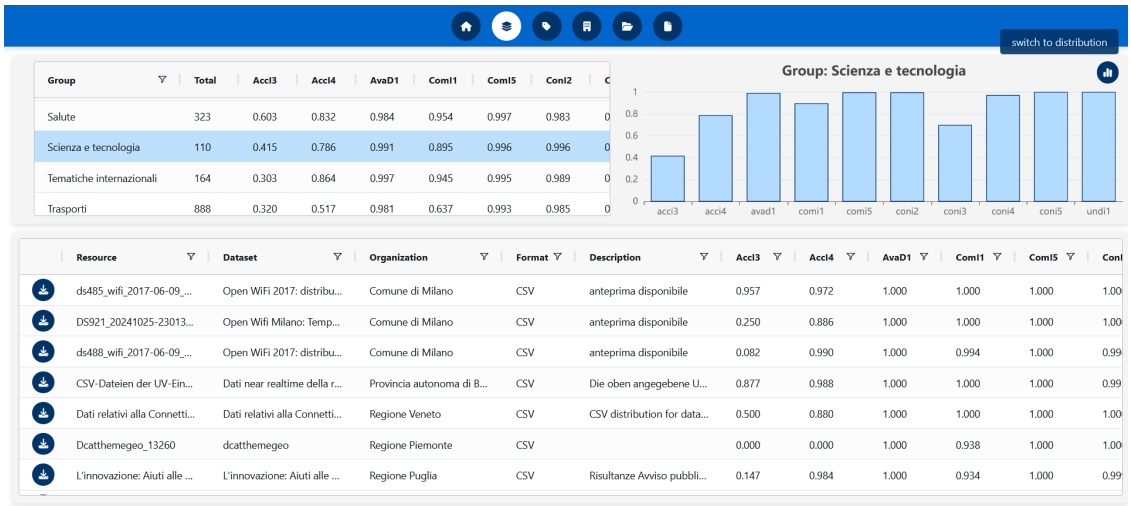

Figure 3.3. User interface dimensions page

The upper section provides the aggregated view for the selected dimension while the lower section provides a detailed view at the level of resource. The dimension's grid on the left contains records of all the possible values for the selected dimension (ex: Group), the total number of resources and the average scores for that value. The selected dimension value (ex: Scienza e tecnologia) in this grid is highlighted in light blue and it commands the data visualized in the other components.

The upper right section offers the graphical presentation of the scores for the selected dimension value in the dimension grid. It is possible to switch between the bar chart and box plot by clicking the button in the upper left corner.

The lower section is synchronized with the dimension grid as well. It shows a detail grid where the records are the metadata and quality scores for the resources corresponding to the selected dimension value. It's possible to download the single resources through the button at the beginning of each record.

The graphs are implemented from the ECharts library while the grids are implemented from the AG Grid library. The grids support sorting, filtering, and column reordering. The selection dimension value is managed in a React state variable whose update is triggered every time the user changes the selected value on the dimension grid. The Resource dimension contains just the detail grid since its the resource is the lowest level of detail that can be reached so aggregating is meaningless. The data used in the Dimension views is retrieved from the /api/group, /api/tag, /api/organization, /api/dataset, /api/resource endpoints.

Chapter 4

Analysis

This chapter presents a massive and detailed analysis of the data quality results measured on the open data datasets collected from the portal. The measures are based on the ISO/IEC 25012 and ISO/IEC 25024 standards and concern the accuracy, availability, completeness, consistency, and understandability of the data. The insights gained from this analysis can help identify strengths and weaknesses in the data management processes and propose new best practices and strategies for improvement.

Measure	Mean	Min	Q1	Median	Q3	Max
Acc-I-3	0.413	0.000	0.030	0.387	0.714	1.000
Acc-I-4-IT-1	0.707	0.000	0.769	0.915	0.971	1.000
Ava-D-2	0.986	1.000	1.000	0.143	1.000	1.000
Com-I-1-IT-1	0.873	0.000	0.833	0.000	1.000	1.000
Com-I-5	0.994	0.000	1.000	1.000	1.000	1.000
Con-I-2-IT-1	0.989	0.000	1.000	1.000	1.000	1.000
Con-I-3	0.596	0.000	0.371	0.604	0.856	1.000
Con-I-3-IT-1	0.947	0.001	1.000	1.000	1.000	1.000
Con-I-5	1.000	1.000	1.000	1.000	1.000	1.000
Und-I-1	1.000	0.778	1.000	1.000	1.000	1.000

Table 4.1. Data quality measure scores

As of 25/10/2024, the portal has registered a total of 63,874 datasets and 206,331 resources. Out of these resources, 53,921 are of CSV format and 22,115 have been measured for data quality. The rest are still to be processed, currently unavailable, or wrongly labeled as CSV.

Starting with accuracy, a mean of 0.413 for **Acc-I-3** indicates that the current algorithm provides only moderate coverage for accuracy. In fact, Acc-I-4-IT-1, the only other measure on accuracy, only calculates outliers for numeric values so the coverage depends on the ratio of numeric values present in each resource. Some resources contain only numeric data (maximal coverage of 1.000) while some others contain none (minimal coverage of 0.000). The resource are very different in this aspect which is also reflected by the high variability of the measure with the IQR being 0.684. Acc-I-3 is not a direct measure of quality but it provides important context with regards to the attention given to accuracy measuring.

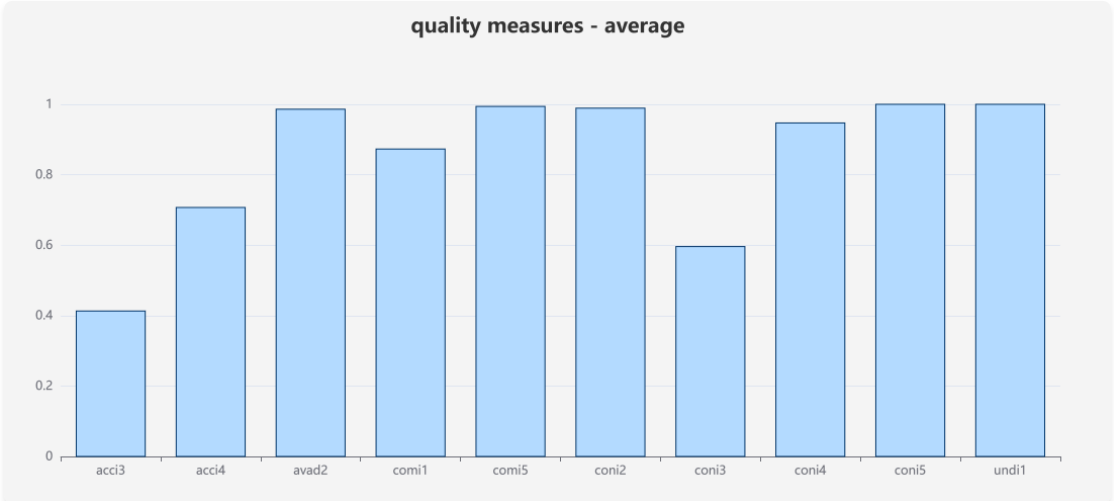

Figure 4.1. Data quality measure scores average

Acc-I-4-IT-1 is characterized by a mean of 0.707 and even higher values for the quartiles which suggest high confidence in the accuracy of numeric data. A high value hints at a low number of outliers present in the data. As explained previously, the score of this measure is to be considered in the context of the coverage calculated by Acc-I-3. A score of 0.000 does not mean that the data is full of outliers, but that the data potentially does not have any numeric values and thus no conclusions can be drawn about its accuracy. Finally, the outliers, when present, have to be validated by a domain expert.

The close to perfect mean and quartile values of **Ava-D-2** demonstrate that availability is strong for most resources. It's worth noting that the data quality measures are calculated for resources that have been found available at least once and were able to be discovered by the tool. There is a considerable number of resources that are constantly unavailable or unprocessable due to being wrongly labeled.

Figure 4.2. Data quality measure scores distribution

Com-I-1-IT-1 exhibits a mean of 0.873 signifying that the majority of the data items are not null values and that the resource files are generally complete. **Com-I-5** has an even higher mean of 0.994 suggesting that there are almost no completely empty records across all resources. Some outliers are present in both measures, generally due to empty resource files, but overall completeness is a strong suit.

Con-I-2-IT-1 and **Con-I-4-IT-1** reflect strong consistency with their good scores. The respective means are 0.989 and 0.947 and the quartiles are all at 1.000. This shows that the data types of the columns and the data structure of the rows is consistent in most of the resources. The few encountered outliers are generally the empty files that have no rows or columns for which consistency can be measured.

Con-I-3-IT-1 measures the risk of duplicated values which can lead to inconsistencies. Any update operations would need to be performed on all the occurrences of the same value in order to avoid inconsistencies. A lower mean of 0.596 for this measure points to only moderate confidence in the absence of duplications and suggests that more attention is required when handling updates in the content of the resources with lower scores.

Con-I-5 is not a direct measure of quality but a measure of the coverage and importance given to consistency. The mean of 1.000 shows that the data values are always measured for consistency which gives more credibility to the evaluation of this quality characteristic as a whole.

Und-I-1 with a perfect score of 1.000 across mean, median, and quartiles demonstrates that the resource files are comprehensible and correctly encoded. This does not mean that understandability is a given in every case as shown by the presence of some outliers and the minimum value of 0.778. It is important to keep tracking this measure since a resource that can't be interpreted is of no use, meaning that any score that is not close to perfect for this measure should be cause of concern.

By using some statistical correlation coefficients, some interesting relationships emerge between the different data quality characteristics. A strong positive correlation between Com-I-5 and Con-I-4-IT-1 indicates that resources with high record completeness are also structurally consistent. These quality characteristics are likely interconnected due to their reliance on robust data entry and validation mechanisms.

Evaluating the overall data quality of the Italian open data some strengths and weaknesses can be pointed out. The high availability and understandability scores show good levels of accessibility and clarity for the open data which is crucial for for this data to be transparent and usable. The completeness and consistency also show encouraging scores reflecting good practices in the preparation and maintenance of the data. However, there are some gaps in the evaluation of accuracy and in the management of duplicate values. The accuracy measurements can be extended to data values other than the numeric ones if a complete and more detailed metadata catalog were to be available. Having more data about the data and its life-cycle can pave the way for the development of more quality measures and a deeper level of analysis.

Chapter 5

Conclusions

5.1 Future Work

After having performed a massive analysis, some opportunities for future improvement emerge, both for the tool itself and for the open data portal.

Expansion of supported data quality characteristics: The tool's quality algorithm evaluates accuracy, availability, completeness, consistency, and understandability. By identifying appropriate measures and with some additional support from the portal metadata, it would be possible to expand to other quality characteristics such as currentness, efficiency, precision. For example, the Cur-I-1 measure on currentness calculates the degree to which the data is being updated with the required frequency. The portal provides among the dataset metadata a field called "frequency" but it rarely contains a non null or valid value.

Expansion of supported data formats: The CSV format is the most common one in the portal, making up to 25% of the total resources. However, there are other commonly used format such as JSON with 15%, XML with 5%, Excel with 4%, etc. The main challenge in normalizing these formats is their variable structure. A combination of more advanced libraries for data normalization and metadata that describe the resource structure can help with the integration of other formats.

Data quality monitoring alerts and reports: Given the automated and continuous processing of the data, an alerting system could be useful for notifying anomalies or data quality degradation to ensure timely intervention. Another convenient feature could be the generation of data quality reports in tabular or image formats which can be used to inform stakeholders or data owners about the results of data quality analysis.

5.2 Final Remarks

This thesis brings attention to the potential of utilizing open data and the importance of data quality. If data is to be used in decision-making processes or analyzed to gain some insight, it is crucial that it maintains a high quality standard. To this end, a tool that combines an automated process pipeline and a user-friendly data analysis interface has been developed.

The ISO/IEC 25012 and ISO/IEC 25024 standards are the basis of the tool's generalized data quality algorithm. The measures on quality characteristics such as accuracy, availability, completeness, consistency, and understandability help in identifying strengths and weaknesses in the management of data during its entire life-cycle. Furthermore, the data discovery, data normalization, and data quality workflows eliminate any required manual effort in the preparation phase and guarantee high reliability in handling large volumes of diverse datasets. The intuitive interface allows users to explore data and metadata and analyze quality results.

The modular architecture allows for the easy integration of new datasets, measures, or features. The entire system is robust and can operate without disruptions thanks to its reliable and continuous logging and error-handling mechanisms. Looking forward, this work lays a foundation for future developments such as supporting additional data formats or integrating advanced machine learning techniques for predictive quality assessment.

In conclusion, the analysis of the data quality results reveals both promising signs and areas for improvement, underscoring the need for continuous monitoring and refinement. By providing a scalable, automated, and user-focused solution, this tool makes a significant contribution to advancing open data practices, ensuring that the promise of transparency and innovation is matched by the quality of the data itself.

Bibliography

[1] Open Definition. (n.d.). The Open Definition. Consulted 10/11/2024, at https://opendefinition.org

[2] European Union. (n.d.). The official portal for European data. Consulted 10/11/2024, at https://data.europa.eu/en/dataeuropa-academy/what-open-data

[3] International Organization for Standardization. (2008). ISO/IEC 25012:2008(E). Software engineering - Software product quality requirements and evaluation (SQuaRE) - Data quality model.

[4] Agenzia per l'Italia Digitale. (n.d.). I dati aperti della pubblica amministrazione. Consulted 12/11/2024, at https://www.dati.gov.it/chi-siamo

[5] International Organization for Standardization. (2015). ISO/IEC 25024:2015(E). Systems and software engineering - Systems and software quality requirements and evaluation (SQuaRE) - Measurement of data quality.

[6] Kristi Gjerko. (2024). A massive analysis of Italian open data quality. Consulted 14/11/2024, at https://github.com/kristigjerko301101/open-data-quality

[7] Marco Rondina. (2023). Experience: Bridging Data Measurement and Ethical Challenges with Extended Data Briefs. Consulted 14/11/2024, at https://github.com/RondinaMR/data-qbd-framework