POLITECNICO DI MILANO

Corso di Laurea Magistrale in Computer Science and Engineering

Scuola di Ingegneria Industriale e dell’Informazione



**A flexible approach to data exploration for big data analysis results**

Relatore: Prof.ssa Cinzia Cappiello

Tesi di Laurea di:

Pavel Gichevski

Matricola : 838627

# Abstract

This work is realized to propose an implementation of a Data Exploration Tool that explores results of a Data Quality as a Service module. The module is placed in the architecture of the EuBRA-BIGSEA project where Politecnico di Milano enters as a collaboration partner. In general, the project aims to develop a set of cloud services impowering Big Data Analytics to ease the development of massive data processing applications.

Data quality in big data has a huge impact in decision making and so it requires understanding of the quality analysis results. To understand the results of a quality analysis we need to explore them and capture their real quality value. Generating value from big data is probably the most important part of a quality analysis because the user needs to make a final decision if the data quality scores are good enough to be used in next computations and statistics software’s, or the data should pertain a new quality analysis with modified settings.

The results of a data quality analysis can be improved varying two things: the data object of interest, and the measures that perform the evaluation. The re definition of the data object leads to variation of the quality results since each aspect of the data source can be evaluated over set of measures, and vice versa, each configuration of the measures gives different scores. Said that, this thesis founds its place as a support of the Quality Service in the BIGSEA project.  
The idea is to allow professionals and newcomers in the field of quality analysis in Big Data a way of exploring and learning the process of analysis by a guided UI. Besides, the UI will provide a set of features that will allow sorting, searching, custom filtering, and even exporting data. This will save the user iterative usage of the quality service, meaning less processing power.

**List of Figures**

Table of Contents

[Abstract 2](#_Toc509526176)

[chapter 1 5](#_Toc509526177)

[1.1 General 5](#_Toc509526178)

[chapter 2 9](#_Toc509526179)

[chapter 3 25](#_Toc509526180)

[chapter 4 40](#_Toc509526181)

[chapter 5 68](#_Toc509526182)

# chapter 1

**Introduction**

## 1.1 General

Since the beginning of the 21st century and the massive internet coverage that the world now has, we are witnesses of big changes in the technologies. Not so long ago it was possible to read the news on our phones, to search for job opportunities, or even to chat with someone over the internet and be “online” without interruption. With this change every IT company oriented their services and adapted to the new era. This lead to the Internet of Things, to Cloud computing, and AI. As the technologies started being accepted by the world a lot of information started to be accumulated, processed, and even bargained on the market. Just the fact that we carry over 100GB of data on over phone tells us how the measurements and processing power changed with the years.

All this changes in the field of data and the IOT attracted the attention of a lot of leading companies and educational institutions to invest in its further processing. Based on this we saw a lot of discoveries in terms of medicine, finance, statistics, weather, etc. Later, it became a trend to use Big data in order to understand the customers, to predict their logic, to sell them the things they exactly need.

It became possible to collect data from various sources and contents in instants of seconds and that lead to even bigger sets of information. However, it’s not about quantity but the quality, and so it is about the data. The data must have a high-quality in order to bring value, which can later be used in decision-making. This requires an assessment framework and assessment process oriented on Big Data because once the size changes we cannot apply the same analytical processes we did on small sets of data. This will be the starting point of the thesis and an introduction to our topic which covers a specific part of the Big Data quality.

## 1.2 EU-BRA Project and Politecnico Di Milano

EUBra-BIGSEA is a project funded in the third coordinated call Europe – Brazil focused on the development of advanced QoS services for Big Data applications, demonstrated in the scope of the Massive Connected Societies.

EUBra-BIGSEA is developing a QoS architecture that will predict resource consumption of Big Data Analytics applications in order to pre-allocate and dynamically adjust virtualized infrastructures. EUBra-BIGSEA will leverage mixed horizontal and vertical elasticity on hybrid container and VM infrastructures to support a rich Data Analytic framework powered by OPHIDIA, COMPSs and Spark, which will be enriched with these capabilities.

EUBra-BIGSEA is an API-oriented project (that is, it is oriented to service and application developers). It will ease the development of Data Analytics application and it will provide improved performance and better usage of resources, as well as a sound framework for implementing privacy policies. EUBra-BIGSEA will be demonstrated on a data-intensive use case on traffic information prediction and recommendation from the municipality data of Curitiba, in Brazil. The final application will target general citizens, who could have a predictive information based on climate conditions and historic data.  Furthermore EUBra-BIGSEA is an international initiative, designed to create and reinforce international cooperation between Europe and Brazil, and will contribute to technology and knowledge exchange between the two.

In particular, the platform has to be able to manage and store different types of data to offer a set of Big Data services.

The services are:

* The data ingestion service that is used to load the data from sensors

and devices distributed over the city.

* The query service that is used to make selections on the stored data

source in order to retrieve information.

* The data quality service that is used to evaluate the quality aspects

of the stored data sources. The produced quality metadata from the

quality analysis will be used by data mining applications, predictive

models and descriptive models.

The data quality as a service of this project is tested in a real scenario. The scenario is the public transportation system of the city of Curitiba, in Brazil. The analyzed data sources are a heterogeneous data sources collection that includes: the registration of the ticket validation of the users, the records related to the localization of the buses in the city, the weather conditions, the events that are extracted from the social networks posts. So, the quality service must deal with many data sources that have different data types, data formats and dimensions.

Using the API’s provided by the service an application tool was developed in Politecnico Di Milano.  
The application offers to the user possibility to set up data quality analysis using the Data Quality Service. After setting up the analysis and getting the result, we should be able to get the most out of them. So, here comes in place the purpose of this thesis. The idea is to let the user either a professional in the field of data analytics or not, to examine the produced results, to understand more for each of the quality dimensions that were used to evaluate different aspects of the data source, to know on which type of data (float, string, datetime, etc.) which dimension can be applied. In details all the features of the data exploration tool will be explained in the following chapters.

## 1.1 Structure of the document

**Chapter 2** State of the Art

**Chapter 3** EU-BRA Data Quality Service

**Chapter 4** Data Exploration Tool

**Chapter 5** Conclusion

# chapter 2

**State of the Art**

## 2.1 Big Data Fundamentals

Big Data is a field dedicated to the analysis, processing, and storage of large collections of data that frequently originate from disparate sources. Big Data solutions and practices are typically required when traditional data analysis, processing and storage technologies and techniques are insufficient. Specifically, Big Data addresses distinct requirements, such as the combining of multiple unrelated datasets, processing of large amounts of unstructured data and harvesting of hidden information in a time-sensitive manner. Although Big Data may appear as a new discipline, it has been developing for years. The management and analysis of large datasets has been a long-standing problem—from labor-intensive approaches of early census efforts to the actuarial science behind the calculations of insurance premiums. Big Data science has evolved from these roots. In addition to traditional analytic approaches based on statistics, Big Data adds newer techniques that leverage computational resources and approaches to execute analytic algorithms. This shift is important as datasets continue to become larger, more diverse, more complex and streaming-centric. While statistical approaches have been used to approximate measures of a population via sampling since Biblical times, advances in computational science have allowed the processing of entire datasets, making such sampling unnecessary.

The analysis of Big Data datasets is an interdisciplinary endeavor that blends mathematics, statistics, computer science and subject matter expertise. This mixture of skillsets and perspectives has led to some confusion as to what comprises the field of Big Data and its analysis, for the response one receives will be dependent upon the perspective of whoever is answering the question. The boundaries of what constitutes a Big Data problem are also changing due to the ever-shifting and advancing landscape of software and hardware technology. This is due to the fact that the definition of Big Data takes into account the impact of the data’s characteristics on the design of the solution environment itself. Thirty years ago, one gigabyte of data could amount to a Big Data problem and require special purpose computing resources. Now, gigabytes of data are commonplace and can be easily transmitted, processed and stored on consumer-oriented devices. Data within Big Data environments generally accumulates from being amassed within the enterprise via applications, sensors and external sources. Data processed by a Big Data solution can be used by enterprise applications directly or can be fed into a data warehouse to enrich existing data there. The results obtained through the processing of Big Data can lead to a wide range of insights and benefits, such as:

* operational optimization
* actionable intelligence
* identification of new markets
* accurate predictions
* fault and fraud detection
* more detailed records
* improved decision-making
* scientific discoveries

Evidently, the applications and potential benefits of Big Data are broad. However, there are numerous issues that need to be considered when adopting Big Data analytics approaches. These issues need to be understood and weighed against anticipated benefits so that informed decisions and plans can be produced.

First to be able to adopt Big Data analytics one needs to know more about the data source, its characteristics, sensors that obtain the data, and possible failures. After that a crucial meaning in the process of data acquisition and processing goes on the architectures used to query, process, and result the data.

### 2.1.1 Big Data Characteristics

For a dataset to be considered Big Data, it must possess one or more characteristics that require accommodation in the solution design and architecture of the analytic environment. Most of these data characteristics were initially identified by Doug Laney in early 2001 when he published an article describing the impact of the volume, velocity and variety of e-commerce data on enterprise data warehouses. To this list, veracity has been added to account for the lower signal-to-noise ratio of unstructured data as compared to structured data sources. Ultimately, the goal is to conduct analysis of the data in such a manner that high-quality results are delivered in a timely manner, which provides optimal value to the enterprise. This section explores the five Big Data characteristics that can be used to help differentiate data categorized as “Big” from other forms of data. The five Big Data traits shown in Figure 1.11 are commonly referred to as the Five Vs:

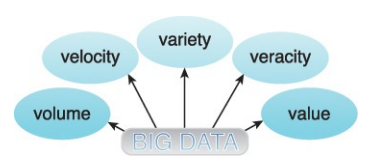


Figure 1: The Five Vs of Big Data

**Volume** The anticipated volume of data that is processed by Big Data solutions is substantial and ever-growing. High data volumes impose distinct data storage and processing demands, as well as additional data preparation, curation and management processes.

Typical data sources that are responsible for generating high data volumes can include:

• online transactions, such as point-of-sale and banking

• scientific and research experiments, such as the Large Hadron Collider and Atacama Large Millimeter/Submillimeter Array telescope

• sensors, such as GPS sensors, RFIDs, smart meters and telematics • social media, such as Facebook and Twitter

**Velocity** in Big Data environments, data can arrive at fast speeds, and enormous datasets can accumulate within very short periods of time. From an enterprise’s point of view, the velocity of data translates into the amount of time it takes for the data to be processed once it enters the enterprise’s perimeter. Coping with the fast inflow of data requires the enterprise to design highly elastic and available data processing solutions and corresponding data storage capabilities. Depending on the data source, velocity may not always be high. For example, MRI scan images are not generated as frequently as log entries from a high-traffic webserver. Data velocity is put into perspective when considering that the following data volume can easily be generated in a given minute: 350,000 tweets, 300 hours of video footage uploaded to YouTube, 171 million emails and 330 GBs of sensor data from a jet engine.

**Variety** Data variety refers to the multiple formats and types of data that need to be supported by Big Data solutions. Data variety brings challenges for enterprises in terms of data integration, transformation, processing, and storage. Representation of data variety include structured data in the form of financial transactions, semi-structured data in the form of emails and unstructured data in the form of images.

**Value** is defined as the usefulness of data for an enterprise. The value characteristic is intuitively related to the veracity characteristic in that the higher the data fidelity, the more value it holds for the business. Value is also dependent on how long data processing takes because analytics results have a shelf-life; for example, a 20-minute delayed stock quote has little to no value for making a trade compared to a quote that is 20 milliseconds old. As demonstrated, value and time are inversely related. The longer it takes for data to be turned into meaningful information, the less value it has for a business. Stale results inhibit the quality and speed of informed decision-making.

The figure below gives brief examples for each of the Vs of Big data.

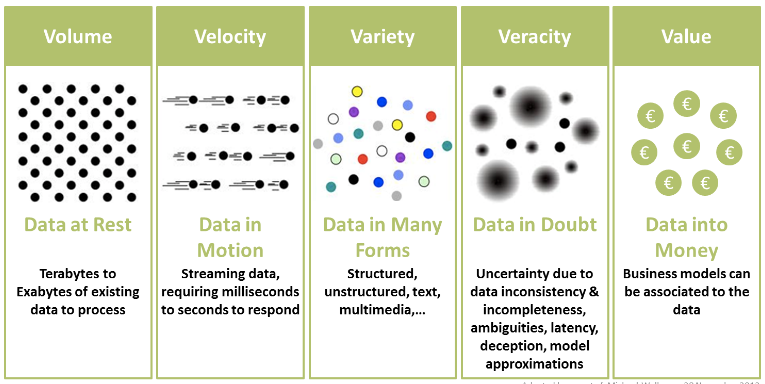


Figure 2Illustration of the Vs of big data

### 2.1.2 Creating Big Values from Big Data

A three-step approach below can help to determine how to create Big Value from Big Data.

* Start with the Right Big Data Store

Matching the business problem or opportunity with the right technology is an important first step. Big Data stores fit into one of several categories: Hadoop (which is a software framework that includes a Big Table clone called HBase), NoSQL (which is subdivided into several more categories, including Key Value Stores, Document Databases, Graph Databases, Big Table Structures, and Caching Data Stores), Analytical Databases (e.g. Infobright, VectorWise, Vertica, Netezza, etc.).

* Add Deep Domain Knowledge

Domain knowledge is the human intelligence that accumulates within a certain practice or process. A “domain” in this sense could be a functional application area (like CRM or Supply-Chain), a vertical industry (like financial services, pharmaceuticals, or energy/utilities), or a specific process (like after-sales support). Domain expertise is necessary to genuinely know which data, from all the possible sources, are valuable and which are not. Domain knowledge is the primary reason the Big Data opportunity requires business unit personnel to lead rather than follow more than ever before.

* Apply the Right Reporting & Analysis Tool

Choosing the right reporting and analysis tool that enables the right overall big data approach (or architecture) is perhaps the most important step. Here finds place the idea of the thesis. Using the Data Quality as a Service we get set of quality results on a set of records that we predefined. Knowing the data object and the dimension measures of quality we can create a bigger value of the data.

## 2.2 Data Quality in Big Data

Data quality is not necessarily data that is devoid of errors. Incorrect data is only one part of the data quality equation. Most experts take a broader perspective. Some experts say data quality involves “consistently meeting knowledge worker and end-customer expectations.” Others say data quality is the fitness or suitability of data to meet business requirements. In any case, most cite several attributes that collectively characterize the quality of data. In order for the analyst to determine the scope of the underlying root causes and to plan the ways that tools can be used to address data quality issues, it is valuable to understand these common data quality attributes.

Figure 3 shown the first five attributes (i.e. Accuracy, Integrity, Consistency, Completeness and Validity) generally intrinsic to the content and structure of data and cover a multitude of sins that we most commonly associate with poor quality data: data entry errors, misapplied business rules, duplicate records, and missing or incorrect data values. But defect-free data is worthless if knowledge workers cannot understand or access the data in a timely manner. The last two attributes (Timeliness and Accessibility) address usability and usefulness, and they are best evaluated by interviewing and surveying business users of the data. Data quality is an essential characteristic that determines the reliability of data for making decisions. High data quality is:

* Complete: All relevant data such as accounts, addresses and relationships for a given customer is linked.
* Accurate: Common data problems like misspellings, typos, and random abbreviations have been cleaned up.
* Available: Required data are accessible on demand; users do not need to search manually for the information.
* Timely: Up-to-date information is readily available to support decisions.

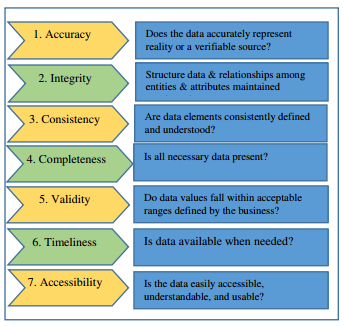


Figure 3 Data Quality Attributes

Amongst others, there are several conditions that contributed to the data quality problem such as lack of validation routines, data valid, but not correct, mismatched syntax, formats, and structures, unexpected changes in source system, spiderweb of interfaces, lack of referential integrity checks, poor system design and data conversion errors. According to TDWI’s Data Quality Survey, almost half of companies (40%) have suffered losses, problems, or costs due to poor quality data and 43% have yet to study the issue. The two most common problems caused by poor quality data are extra time required to reconcile data and loss of credibility in the system or application. Fig. 4 represents the output of 286 respondents who could select multiple answers on the data quality problem.

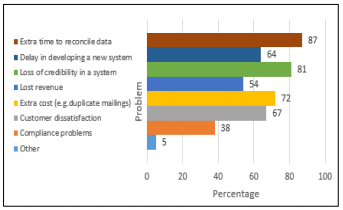


Figure 4:Problems due data quality

### 2.2.1 Data Quality and Measurement

When we speak of measurement we usually think of taking notes of size, amount, or degree of something by using an instrument. But, in the case of Data Quality we think of a process that lets us compare values to a well-known standard, or in other words, to assess the information and draw a conclusion.

A lot of experts define the purpose of data quality assessment: to identify data errors and erroneous data elements and to measure the impact of various data-driven business processes. Both components -- to identify errors and to understand their implications -- are critical. Data quality assessment can be accomplished in different ways, from simple qualitative assessment to detailed quantitative measurement. Assessments can be made based on general knowledge, guiding principles or specific standards. Data can be assessed at the macro level of general content or at the micro level of specific fields or values. The purpose of data quality assessment is to understand the condition of data in relation to expectations or particular purposes or both and to draw a conclusion about whether it meets expectations or satisfies the requirements of particular purposes. This process always implies the need also to understand how effectively data represents the objects, events and concepts it is designed to represent.

The word dimension is used to identify aspects of data that can be measured and through which data's quality can be described and quantified. So, it means that the quality of a data source can be evaluated from different perspectives, and on a different level of granularity. The quality of a data source can be evaluated under different aspects and the level of quality will depend on the set of quality dimensions that we want to use. With different aspects of the data source we mean by exploring different sets of records of a data set that we think will give different quality results. So, each quality dimension will give indication how good or bad that data object (subset of records selected from the whole data source) is.

There are 179 quality dimensions, but in the interest of the thesis we will focus only on 9 of them, since they are used in the Quality Service of the EU-BRA project.

All Data Quality dimensions can be grouped in four categories:

* Intrinsic: the data has a quality on its own. For example, Accuracy is a quality dimension that is intrinsic to data.
* Contextual: highlights the requirement that must be considered within the data source. It must be relevant, timely, complete, and appropriate in terms of amount.
* Representational: focuses on aspects related to the quality of data representation
* Accessibility: considers the accessibility of the data and its level of security

### 2.2.2 Big Data Architectures

As the world becomes more information-driven than ever before, a major challenge has become how to deal with the explosion of data. Traditional frameworks of data management now buckle under the gargantuan volume of today's datasets. Fortunately, a rapidly changing landscape of new technologies is redefining how we work with data at super-massive scale. These technologies demand a new breed of DBAs and infrastructure engineers/developers to manage far more sophisticated systems.

Here is an overview of important technologies to know about for context around big data infrastructure.

### 2.2.2.1 NoSql

Relational databases built around the SQL programming language have long been the top -- and, in many cases, only -- choice of database technologies for organizations. Now, with the emergence of various [NoSQL software platforms](http://www.theserverside.com/news/2240233295/What-matters-most-in-micro-services-REST-and-NoSQL), IT managers and business executives involved in technology decisions have more options on database deployments. Databases support dynamic schema design, offering the potential for increased flexibility, scalability and customization compared to relational software. That makes them a good fit for Web applications, content management systems and other uses involving large amounts of non-uniform data requiring frequent updates and varying field formats. In particular, [NoSQL technologies](http://searchdatamanagement.techtarget.com/ezine/Business-Information/NoSQL-technologies-take-on-rising-tide-of-big-data) are designed with "big data" needs in mind.

The efficiency of NoSQL can be achieved because unlike relational databases that are highly structured, NoSQL databases are unstructured in nature, trading off stringent consistency requirements for speed and agility. NoSQL centers around the concept of distributed databases, where unstructured data may be stored across multiple processing nodes, and often across multiple servers. This distributed architecture allows NoSQL databases to be horizontally scalable; as data continues to explode, just add more hardware to keep up, with no slowdown in performance. The NoSQL distributed database infrastructure has been the solution to handling some of the biggest data warehouses on the planet – i.e. the likes of Google, Amazon, and the CIA.  
A comparison between the scalability of NoSql database vs Traditional Relational Database can be seen in Figure 5.

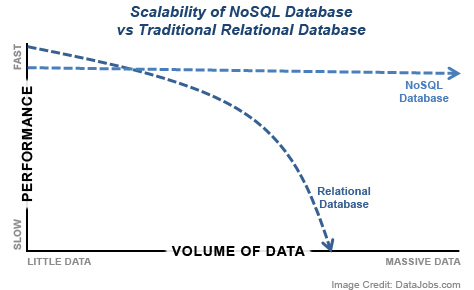


Figure 5 Scalability of NoSQL vs Traditional RDB

### 2.2.2.2 Apache Hadoop

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

* Hadoop Common: The common utilities that support the other Hadoop modules.
* Hadoop Distributed File System (HDFS™): A distributed file system that provides high-throughput access to application data.
* Hadoop YARN: A framework for job scheduling and cluster resource management.
* Hadoop MapReduce: A YARN-based system for parallel processing of large data sets.

A staple of the Hadoop ecosystem is MapReduce, a computational model that basically takes intensive data processes and spreads the computation across a potentially endless number of servers (generally referred to as a Hadoop cluster). It has been a game-changer in supporting the enormous processing needs of big data; a large data procedure which might take 20 hours of processing time on a centralized relational database system, may only take 3 minutes when distributed across a large Hadoop cluster of commodity servers, all processing in parallel.

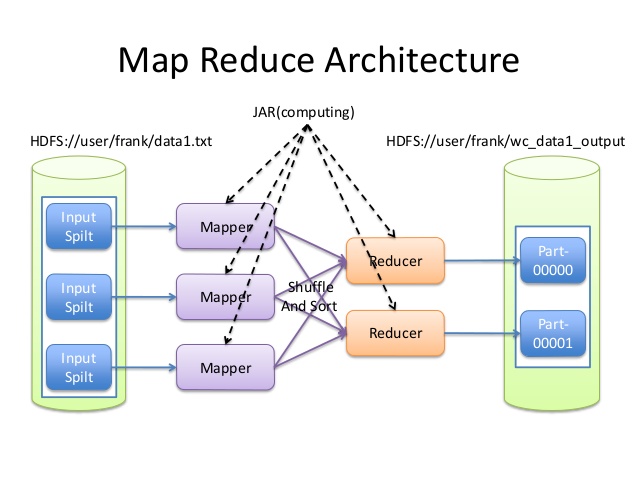


Figure 6 Map Reduce Architecture

### 2.2.2.3 Apache Spark

Apache Spark is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets. With Spark running on Apache Hadoop YARN, developers everywhere can now create applications to exploit Spark’s power, derive insights, and enrich their data science workloads within a single, shared dataset in Hadoop.

The Hadoop YARN-based architecture provides the foundation that enables Spark and other applications to share a common cluster and dataset while ensuring consistent levels of service and response. Spark is now one of many data access engines that work with YARN in HDP.

Apache Spark consists of Spark Core and a set of libraries (Figure 7). The core is the distributed execution engine and the Java, Scala, and Python APIs offer a platform for distributed ETL application development.

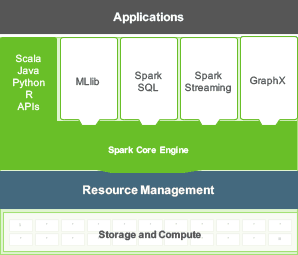


Figure 7 Apache Spark Libraries

Additional libraries, built atop the core, allow diverse workloads for streaming, SQL, and machine learning.

Spark is designed for data science and its abstraction makes data science easier.   Data scientists commonly use machine learning – a set of techniques and algorithms that can learn from data. These algorithms are often iterative, and Spark’s ability to cache the dataset in memory greatly speeds up such iterative data processing, making Spark an ideal processing engine for implementing such algorithms.  
Here are the benefits of Apache and a comparison with Hadoop:

* Speed: Engineered from the bottom-up for performance, Spark can be [100x faster than Hadoop for large scale data processing](https://databricks.com/blog/2014/11/05/spark-officially-sets-a-new-record-in-large-scale-sorting.html) by exploiting in memory computing and other optimizations. Spark is also fast when data is stored on disk, and currently holds the world record for large-scale on-disk sorting.
* Ease of Use: Spark has easy-to-use APIs for operating on large datasets. This includes a collection of over 100 operators for transforming data and familiar data frame APIs for manipulating semi-structured data.
* A Unified Engine: Spark comes packaged with higher-level libraries, including support for SQL queries, streaming data, machine learning and graph processing. These standard libraries increase developer productivity and can be seamlessly combined to create complex workflows.

## 2.3 Conclusion

As big data continues down its path of growth, there is no doubt that these innovative approaches – utilizing NoSQL database architecture and Hadoop software – will be central to allowing companies reach full potential with data. Additionally, this rapid advancement of data technology has sparked a rising demand to hire the next generation of technical geniuses who can build up this powerful infrastructure. The cost of the technology and the talent may not be cheap, but for all of the value that big data is capable of bringing to table, companies are finding that it is a very worthy investment.

Having in mind all the data quality dimension mention before, and all the technologies that can help us perform analysis over Big Data there is still a gap in the literature because there is no appropriate method to assess the Quality of Big Data.

Taking part in the EU-BRA project, Politecnico Di Milano proposed a method that will address this issue and regard the Quality of Big Data. The Quality Dimensions are being reinterpreted and redefined based on the type of considered data and new metrics take into account the increased complexity generated by the volume, variety and velocity challenges.   
The proposed method was implemented as Data Quality Service and was tested on data sources offered on the cloud platform of the EU-BRA project. Also, an application that offers user interface was created based on the API’s that the service is providing. All with one purpose, to set up a quality analysis on Big Data which will allow further understandings when it comes to quality analysis as the data grows bigger and bigger.

The implementation of the proposed model executes the computation of the quality dimensions using the Spark framework that allows to split the computation of the code over several nodes in order to speed up the computation time of the quality analysis. In Data Mining application the quality of the obtained results is important. One of the factors that influences the quality of the output produced by a data mining application is the quality of the analyzed data object. The implementation of the proposed model will be used to verify the fact that the computation of a generic data mining application over a data object filtered out by some "noise" records, will improve the quality of the results.

In the following chapters we will describe the proposed method for quality analysis and the Quality service in details. After that the thesis concentrates on the data exploration tool that will help the user to better understand the results and then make it easier to re-formulate another quality analysis by varying the quality dimensions or the analyzed data object. With other words, we want to address bigger value of the data quality results.

# chapter 3

**EU-BRA Data Quality Service**

## 3.1 EUBra-BIGSEA Project

EUBra-BIGSEA project aims to develop a cloud platform for Big Data Management and exploitation. It is funded in the third coordinated call Europe – Brazil, focused on the development of advanced QoS services for Big Data applications, demonstrated in the scope of the Massive Connected Societies.  The idea of the project is to develop architecture that will predict resource consumption of Big Data Analytics applications in order to pre-allocate and dynamically adjust virtualized infrastructures. It leverages mixed horizontal and vertical elasticity on hybrid container and VM infrastructures to support Data Analytic framework powered by OPHIDIA, COMPSs, and Spark. The integrated technologies should support:

* Fast data analysis over continuous streams
* Data mining and machine learning
* OLAP-based Big Data analytics

.

The project is API oriented, which means it is oriented to service and application developers. In order to test the services and in the same time to offer the services the project is being tested in a real user scenario in Curitiba, Brazil.

The Data Quality Service is in charge to provide a quality analysis over a data sources that provide information about GPS location of local buses and ticket usage in the same buses.

## 3.2 Model Architecture of the Data Quality Service

The data quality service provides additional information about the data sources saved on the platform. We can distinguish the information in two groups:

* Data Quality Profiling results related to data that is in the data source, such as value ranges, and uniqueness degree of each attribute and number of represented objects.
* Data Quality Assessment results based on the performed quality analysis over the data source, obtained by evaluating specific data quality dimensions.

So, the Data Quality Service is composed of two modules: Profiling module and Assessment module. The first one executes periodically over the whole data source in order to produce general information. This information is being used as input in the second module. For example, when setting the Quality Dimensions for the analysis, the Assessment module uses the metadata of the first module, so it can know which data types can be used for each quality dimension. In case of adding a new attribute in the data source it is crucial to re-run the Profiling module, so all the information is up to date.

The Assessment module is run on demand of users or applications that specify the details of the quality analysis which is being run on the service. Also, the results of the assessment module are kept as metadata for a further analysis.

In the figure bellow we can see the global architecture of the model.

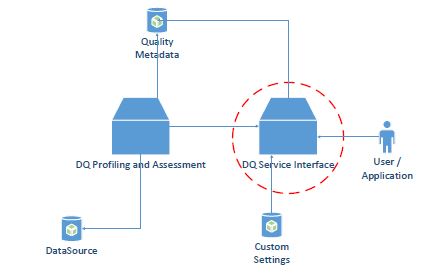


Figure 8 Architecture of the Quality Service

The Data Quality Service is in charge to provide a descriptive view of the quality of the sources with the aim to support the analytics applications in understanding which are the relevant and useful data to consider in more advanced analyses. Once the user specifies the sources of interest the Data Quality Service interface visualizes in the Quality Metadata Repository all the general information about the data, which is the number of records contained in the data source, data ranges of values that each attribute can have. After the data source registration phase the user/application sets up a quality analysis over it. The Assessment module also has access to the Quality Metadata Repository, so the service interface will be familiar with any restrictions on the data source (for example allowed data types for each dimension).

Once the user sets up the quality analysis, all the settings are being saved in the Custom Settings Repository. The purpose is to keep the quality analysis settings for each quality analysis for a possible re computation and modification on the data object or varying values on the dimension set.

The quality request set by the user/application is submitted to the Data Quality Assessment module where all the computations specified in the request take place. Once the results are ready they are saved in the Quality Metadata Repository. The quality information of the selected object can let the user, or any data mining application be aware of the quality of the input data, so it can be useful to consider in more advanced analyses.

To make a brief recap, the Quality Metadata Repository stores metadata both from the results of the Profiling and Assessment Module, and metadata used to define parameters that will be selected using the available interface of the Service.

Parameters are being inserted by the user/app when interacting with the service interface. As far we thought of the user as a professional in the field of data analytics, but the service is also providing an applicable interface for semi-professional users, so it can make the data analysis process more user friendly. For example, the user can select a list of data mining applications in order to set up a quality analysis based on the quality dimensions that the selected data mining application requires. Also, the service is providing an interface where the user can choose from a group of quality dimensions instead of choosing all dimension one by one.

## 3.1 Data Quality Service Interface

The user interacts with the Data Quality Service Interface using a set of Application Programming Interfaces(API). An API allows a user to manipulate resource remotely using communication protocols. The service provides many API’s. Each of them is defining a portion of a quality analysis defined by a user/application. The whole set of communication between the Service and the user is saved into a configuration file. The configuration file keeps all the details of the configuration defined by the user/application and it will be saved in the Custom Settings Repository, from where it is passed to the Data Quality Assessment module.

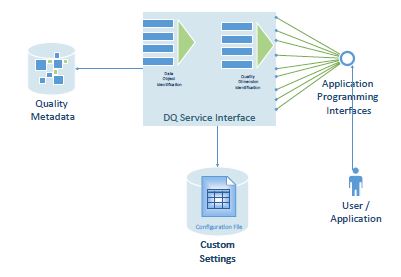


Figure 9 Service - User/app communication with API's

When making a quality analysis over a data source the user/app should enter information about the part of the source that is going to be exploited. This is the process of selecting a Data Object of interest and it forms one of the two main nodes of the service: Data Object Identification Module. Depending on the level of granularity the user is interested to explore, there are three submodules of parameters that are composing this module:

* Data Source Selection Module: selection of a data source (for example it can be the one that keeps track of the BusGPS, or the other with User-ticket information)
* Data Attribute Selection Module: selecting attributes of interest from the data source (for example we can select the code number of the bus only)
* Data Value Selection Module: selecting data values for each of the selected attributes. We can either select a range of values, or specify which values exactly are in our interest.

The module where the service offers the user an interface to select the quality dimensions that will be computed over the previously defined object is Quality Dimension Identification Module. It is composed of four other submodules that will help user either professionals or not to set up a quality analysis. The submodules are the following one:

* Automatic Dimension Identification Interface: it is mainly used by the unexperienced data analysts that want to perform a quality testing on the data source, but they don’t have deeper knowledge in the quality dimensions.
* Focus Category Dimension Identification Interface: Used to set up a quality analysis by selecting a group of dimensions. So, the user has to choose one of the predefined groups which the service determines by reading the profiling data from the Quality Metadata Repository.
* Data Mining Dimension Identification Interface: It proposes to the user a set of data mining applications. If the user knows which data mining application is going to use afterwards than it is a good idea to choose this option.
* Custom Dimension Identification Interface: It is used in the case when the user manually enters each quality dimension that will be computed over the data object. This is mostly used by data analysis professionals.

### 3.1.1 Data Object Identification

The process of selecting the data object has a big importance for the data analysts because it helps them understand interesting information about the whole data source, and usually that’s why the data quality analysis is being re computed many times. If there is no selected portion of interest the whole data object is selected automatically, but this can be risky since the results of the analysis can return an unexpected value which can cause confusion.

When interacting with the Data Quality Interface, the user/application can define the data object in three granularities:

* Source schema selection - The user is obligated to select the source of interest, because without any source the quality analysis cannot continue. Or, alternatively it is also possible to select more than one data source, in a case the user wants to compare the results over the same quality analysis.
* Attribute schema selection - For each data source there is a set of attributes that can be selected by the user. Also, the user is allowed to select multiple attributes of the data source. This will help to gather more knowledge for the data source. This is a crucial process when building the configuration file because the service can check the metadata of the Profiling module and get to know which dimensions can be computed over the selected attributes.
* Value schema selection - The set of attributes selected in the Attribute Schema Selection allows the user to select values for each of them. Without any attribute selected it is not possible to select values, and the quality analysis will be done only on data source level.

There are two possible way to select values. One way is to select a set of values which we are familiar with and we want to observe all quality dimension only on them. The second way is to select a range of values.

### 3.1.2 Quality Dimension Identification

After selecting the data object of interest in the first module, the Data Quality Service provides interface for selecting the Quality Dimension Identification node. This is the most important module since it shapes the quality analysis the user wants to perform over the data object. Each of the selected submodules in a combination with the previously selected Data Object, gives a different quality analysis results.

The identification of quality dimension and theirs level of granularity can differ from user to user, because some of them are more skilled than the others. So, the interface provides step by step dimension identification. The four submodules of the interface are grouped in two groups (Figure 10).

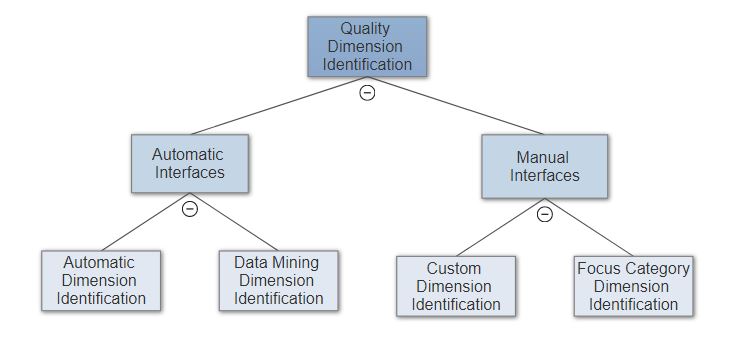


Figure 10 Quality Dimension Identification

#### 3.1.2.1 Automatic Interfaces

Assuming the user that performs the analysis is not skilled and it is only interested in some generic quality indicators about the data object that was firstly selected. With the selection of the automatic interface the user is helped for the setup of the quality dimensions.

* Automatic Dimension Identification: Knowing the attribute that the user previously selected, the data quality service will infer all the available dimensions that can be set to make an analysis. There is need of some metadata parameters of level 2, but in this case, they are already stated by the administration. The level of granularity in which the user can explore the results is the lowest one. It means that the quality dimensions will refer only to the data source.
* Data Mining Dimension Identification: The user chooses a data mining application that will be run over the data object. There are three levels of granularity on which the selected data object can be analyzed, from which the user has to choose only one level of detail on which the quality analysis will be computed.

If the level of detail is “low” it means that each quality dimension will be computed over the data object without giving any attribute quality indicators.

If the level of detail is “medium” it means that the service will produce a quality evaluation over each attribute of the selected data object.

Leve of detail “high” means that the service will produce quality evaluation for each value of the selected attributes from the data object.

#### 3.1.2.2 Manual Interfaces

Assuming the user that performs the quality is an experienced one. The interface offers two possibilities:

* Focus Category Dimension Identification: it is an interface that offers to simplify the setting up of the quality dimensions. Basically, all the dimension are grouped in categories where they have something in common. Each group contains dimensions that are exploiting the same aspect of a data object. The metadata of level 2 Is automatically set by the model.
* Custom Dimension Identification: In this interface the user specifies all the quality dimension on its own, supposing that is an expert and knows which aspect of the data source will be compute by each dimension. In case that some of the quality dimensions require an additional parameter, the user has to provide it on its own (for example the user has to define volatility when choosing timeliness as a quality dimension).

In both manual interfaces the user is required to specify the level of detail on which each quality dimension will process the analysis (global, attribute, value).

## 3.3 Data Quality Service Implementation

In this section we can see how the architecture of the proposed method for a Data Quality Service was implemented, which technologies were used, and API communication with user/app.

### 3.3.1 Selection of Dimensions

The quality evaluation of a data object can be performed over many quality dimensions. The Data Quality Service implements 9 dimensions:

#### 3.3.1.1 Accuracy

It is defined as a closeness between a data value V and a data value V0. Basically, it is the ratio between the two values, letting us know how close is the data value of exploration to the one that is considered as a real-life phenomenon. Returns two indicators:

* Static Accuracy: if a value is in the admirable interval or not
* Dynamic Accuracy: Represents how the value is close with respect to the mean

It is represented as a linear value ranging from 0 to 1.

Granularities: global, attribute, value

Allowed data type: float

#### 3.3.1.2 Completeness

It is used to estimate how many items in the data collection are included in the expected item set. This dimension can be viewed from many perspectives and degrees of granularities which leads to different dimension metrics:

* Completeness Missing: measuring the number of missing/or present values in a data set

It is represented as a linear value ranging from 0 to 1.

Granularities: global, attribute, value

Allowed data types: float, string, datetime

* Frequency Completeness: derived as a ratio between effective frequency of a value Vi measured over the data object represented by parameter Fi, and expected frequency of a value Vi over the data object represented by Hi

It is represented as a linear value ranging from 0 to 1

Granularities: global, attribute, value

Allowed data types: datetime

* Population Completeness: defining how many instances of the data source are included in the data object. If all instances are present, the Completeness population value is 1. Otherwise, the value can be between 0 and 1.

Granularity: attribute

Allowed data types: float, string

#### 3.3.1.3 Consistency

It is a metric that helps the user to understand if there is a logical association between records of the selected data object. It is calculated with the following ratio:

It is represented as a linear value ranging from 0 to 1

Granularities: attribute, value

Allowed data types: all

#### 3.3.1.4 Precision

It is defined as the degree to which repeated measurements show the same or

similar results, and it means that all the values should be equally distributed

and not too distant between each other. Returns three output data related to the analyzed value:

* Precision of the value: how the aggregated values are close to each other
* Association mean of the values
* Standard deviation: In the literature the standard deviation is often used to assess the dispersion of the data, leaving the interpretation of the obtained value to the application or user that requested the analysis, but in order to access the Precision dimension of an entire source, a value between 0 and 1 is returned.

Granularities: global, attribute, value

Allowed data types: float

#### 3.3.1.5 Timeliness

It is defined as “how current the data in the data source is”. Of course, it can be derived if only we have information when the data was created. It is calculated with the following ratio:

Where, the currency is defined as deliveryTime – inputTime in which the first refers to when the data is delivered to the user, and the second refers to when the data is received by the system. Volatility gives information for how long the data is usable.

It is represented as a linear value ranging from 0 to 1

Granularities: global, attribute, value

Allowed data types: timestamp

#### 3.3.1.6 Distinctness

Used in defining how much records in the data is unique. It can be calculated over the whole data source, attributes, or for each value. It is calculated by removing duplicate elements.

It is represented as a linear value ranging from 0 to 1

Granularities: global, attribute

Allowed data types: float, string

#### 3.3.1.7 Volume

It is defined as “number of instances in a selected data object”. So, basically it represents count of each record that forms the data object from the data source. In example if our data source keeps names of users, by performing quality analysis with Volume as a measurement dimension we can get the count for each name in the data source.

It is represented as a linear value ranging from 0 to 1

Granularities: global, value

Allowed data types: float, string

### 3.3.2 Data Quality Service Interface Implementation

The Data Quality Service Interface is implemented as a collection of API’s that divide the work by modules. The service is developed using Jersey RESTful Web Services. The framework itself is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation.

Jersey framework is more than the JAX-RS Reference Implementation. Jersey provides its own [API](https://jersey.github.io/apidocs/latest/jersey/index.html) that extend the JAX-RS toolkit with additional features and utilities to further simplify RESTful service and client development. Jersey also exposes numerous extension APIs so that developers may extend Jersey to best suit their needs.

There are two types of resources offered by the service, one to define the data object and another to set up the quality analysis. Each of the resources is defined with an URI. Using HTTP, the request can be GET, POST, PUT, and DELETE. The responses to the API can be in XML, HTML, or JSON format.

* + 1. Data Quality Service Configuration

When setting up a quality analysis over the Quality Service the user has to choose a data object of interest and to set up quality dimensions that will be computed during the analysis. The APIs exposed by the service allow a user to develop a standalone application, a web or phone application in order to set up the analysis. The idea is to invoke the APIs in such a way that a configuration file will be created to keep the analysis request. Later, according to the configuration file and the profiling metadata that is kept, the quality service can perform the analysis and produce results.

The configuration file is built during the interaction between the user and the service and its identified by a ticket number that the user receives when it starts the interaction. Later, when the configuration file is saved and the user/application makes a request for the analysis to start, the assessment module of the service begins the analysis. Bellow we can see an example of a configuration file used by the service.

/tmp/doc1

/tmp/assessment/test

CODLINHA;CODVEICULO;DATAUTILIZACAO;NUMEROCARTAO;NOMELINHA

CODLINHA,CODVEICULO;CODLINHA;CODVEICULO

;;;;

;;;;

Volume;Precision;Accuracy;Completeness\_Missing;Distinctness;Timeliness;Completeness\_Frequency;Completeness\_Population;Consistency

8760;

0002075900;

1000;

CODLINHA,CODVEICULO:NUMEROCARTAO;CODLINHA:CODVEICULO

global,value;global,attribute,value;global,attribute,value;global,attribute,value;global,attribute;global,attribute,value;global,attribute,value;attribute;attribute,value

0

Figure: Configuration File as input in the quality service

Independent of the application type that invokes the APIs offered by the service the configuration file will be unchanged. In the section bellow we will give a brief explanation of the configuration file.

* 1. Conclusion

The proposed Data Quality as a Service Module took place in the EUBra-BIGSEA project. The Profiling and the Assessment module are scalable because are developed using a Spark code. They are being executed over a number of nodes depending on the volume of the object.

It found good acceptance because not only it offers a data quality over Big Data, but proposes a redefinition of quality dimension. The model proposes an adaptive approach that based on the portion of the data source that needs to be analyzed, helps the user in the definition of quality analysis that can be computed over the data object. It is adaptive because for each change of the data object the quality analysis will change.

To capture the characteristic of a data object we must explore the results of the quality analysis. As the final step of the quality analysis it is crucial to understand the results, so we can proceed by performing a preprocessing task before making available the data object, or in the other case to make another analysis quality that is more focused on a specific aspect of the data object. We can improve the results obtained of the analysis by changing the data object (for example we are interested only in one attribute, or in other case we want to limit values on some of the attributes), or set up different quality dimension to evaluate the same data object.

By varying the quality dimensions, we can also even change the level of detail we want the analysis to take place on. This will make the quality of the results to increase.

In the next chapter we will go in detail over the proposed data exploration tool that helps the users examining the produced results.

# chapter 4

**A methodology for quality results exploration**

To understand the results of a quality analysis we need to explore the results and obtain a real quality value of those. This is probably the most important part of a quality analysis because the user needs to make a final decision if the data quality scores are good enough to be used in next computations and statistics software’s, or the data should pertain a new quality analysis with modified settings.

The results of a data quality analysis can be improved varying two things: the data object of interest, and the configuration applied to the quality dimensions offered by the service.  
The group of quality dimension on which a user wants to perform an analysis can always be manipulated, as it is possible to select the level of details on each of these dimensions. As stated in Chapter 3, the configuration of the service allows the user to manually pick quality dimensions and their level of detail. By detail we mean the granularity level with which the quality dimension will produce an indicator of quality about the data object. This change in granularity level can lead to improved or worsened results.

The re definition of the data object also leads to variation of the quality results since each aspect of the data source can be evaluated over set of dimensions. Having in mind these two factors that are shaping the quality analysis we must be able to understand how the work together and which aspects of the results are affected. In order to understand that easily the need of a data exploration tool emerged. Such a tool will be able to offer the user the possible variations that can be done in order to improve the results.

In this chapter we will discuss main ideas and decisions that are behind the thesis. First in 4.1 we will briefly introduce the outcomes of a quality analysis after the assessment in the QoS service in Big Sea. In section 4.2 we will narrow down the need for a data exploration tool as a support component of the Data Quality Service and application tools depending on the service. Then, in 4.3, we will describe the model of such a tool. In 4.4 and 4.5 we will go in details about the technologies used when implementing the tool and its complexity and working logic accordingly.

4.1 Assessment Results from the EU-Bra Data Quality Service

As it was mentioned in the chapter before, when setting up a quality analysis over the Quality Service the user has to choose a data object of interest and to set up quality dimensions that will be computed during the analysis. The APIs exposed by the service allow a user to develop a standalone application, a web or phone application in order to set up the analysis. The idea was to invoke the APIs in such a way that a configuration file will be created to keep the analysis request. Later, according to the configuration file and the profiling metadata that is kept, the quality service can perform the analysis and produce results.

All the experimental results obtained by the service are over data sources available by the Eu -Bra Bigsea Project. For our interest, in the following section we will concentrate only on experimental results regarding a data source for the public transportation system of the city of Curitiba, Brazil. The source that we will take into inspection in particular in this chapter is related to ticket validations performed by users on the public buses in the city. The available information in the data source is identified by:

* CODLINHA: code of a bus line
* NOMELINJA: name of a bus line
* CODVEICULO: code of the vehicle assigned to the bus line
* NUMEROCARTAO: number of the user card that makes the validation
* DATAUTILIZICAO: date of validation of the ticket

In the figure bellow we can see how the information is saved in the data source:

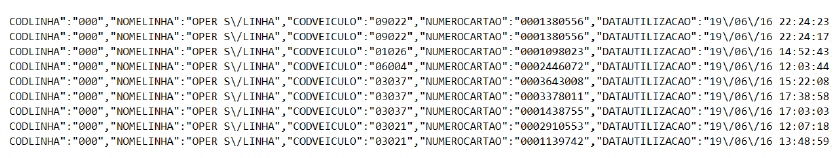


Figure 11: instance of user validation data source

Once the Quality Service finishes with the analysis it saves the results in a directory named “DQAssessment\_results”. The output produced is composed of different objects, where each object represents a quality dimension calculated over an aspect of the data source.

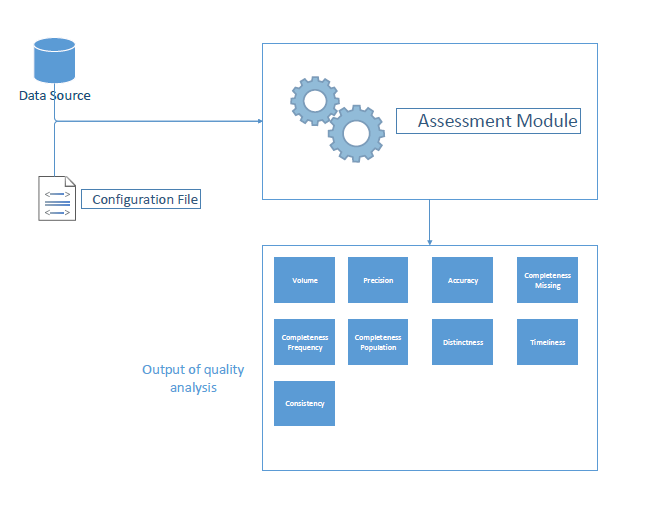


Figure 12: Output of an analysis

When exploring the results of a quality dimension the user can see the evaluation of the dimension on different levels of granularity. The level of granularity can be selected when setting up the analysis and of course if the data object allows such inspection. In the implementation of the Eu-Bra Bigsea project it is possible to inspect three levels of granularity, so the results will be grouped accordingly:

* Global analysis results
* Attribute analysis results
* Value analysis results (for each attribute value of the data source)

Depending the level of granularity selected from the user, the quality service saves the output results in JSON files according quality dimensions. The only exception is the global granularity results file, it is a single file which contains the final results of each quality dimension selected over the whole set of data.

In the figure bellow we can see how the quality service saves the output results in a case where the user requested for all three levels of granularities over all available quality dimensions:

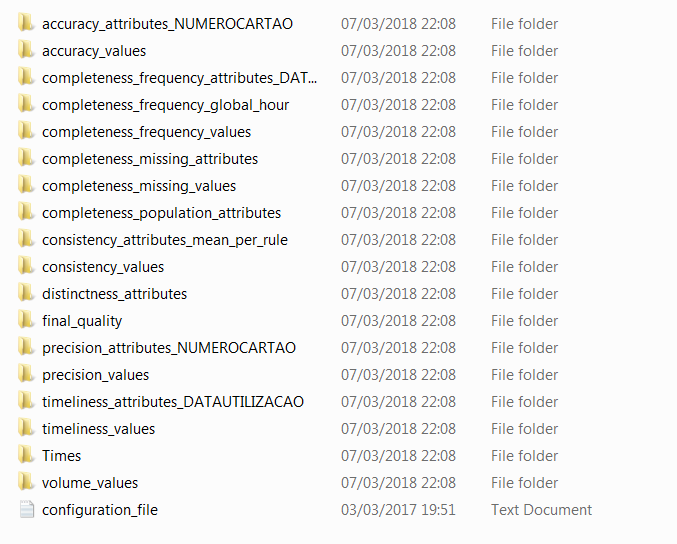


Figure 13: DQAssessment results

From the directory above we can see that the results are saved in three types of folders according the level of granularity.

4.1.1 Global granularity results

“final\_quality” contains a JSON file with evaluation of all quality dimensions of selection over the data source.

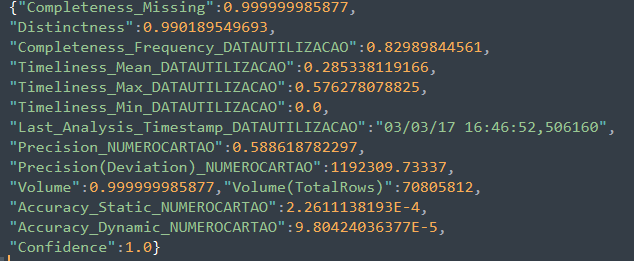


Figure 14: global granularity evaluation

4.1.2 Attribute granularity results

Folders containing “attribute” represent attribute level of granularity. For each dimension the service creates a JSON file with evaluation over each attribute of selection. Of course, not all dimension can be evaluated over attribute level of granularity.

In the examples bellow we can see how consistency and distinctness are evaluated over CODLINHA and CODVEICULO and CODLINHA\_CODVEICULO since they were selected as key attributes by the user when setting up the analysis.

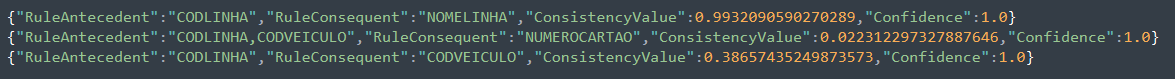


Figure 15:Attribute granularity evaluated over Consistency

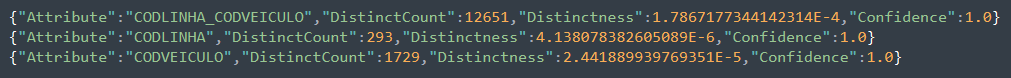


Figure 16: Attribute granularity evaluated over Distinctenss

4.1.3 Value granularity results

Folders containing “value” represent value level of granularity. For each dimension the service creates a JSON file with evaluation over each value of each aggregation attribute. For example, if the aggregation attributes are CODLINHA and CODVEICULO, the file will contain results for each row of records of that attribute. Of course, not all dimension can be evaluated over value level of granularity. In the set in the figure bellow we can see how completeness frequency is calculated over each CODLINHA record in the data source.

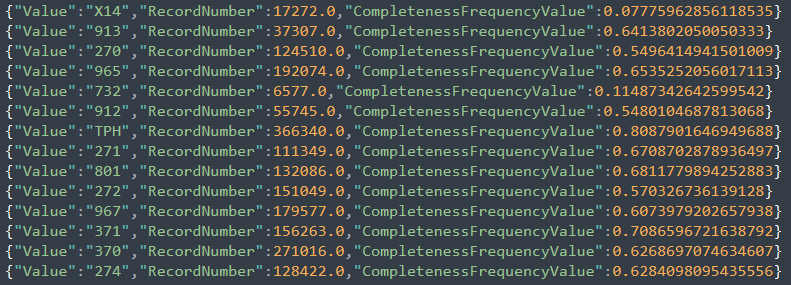


Figure 17: Value level of granularity over CODLINHA and completeness frequency as evaluation dimension

**4.1.4 Variations in the output results**

We have seen an example of an output folder that covers one of the most complex scenarios that can occur when setting up an analysis over the service, since it inspects all three levels of granularity. The set of results will change depending the configuration file and the request of the user. In a case where the user request for only global evaluation of the data source the service will provide one JSON final quality file. In a case where the user wants to inspect only attribute, or value level of granularity, the service will give only that information as an output.

**4.1.4.1 Service constraints**

A notable fact for the upcoming sections of the thesis is that the quality service is offering a selection of quality dimensions and each of them have different constraints over data type and granularity inspection. This must be taken as an important point when modeling a data exploration tool. For example, Value as a quality dimension can’t be explored on attribute level of granularity, and Completeness Population can only be explored on attribute level of granularity.

**4.1.4.2 Additional features**

Beside the mentioned possibilities, the service is built in such a way so it can offer predefined settings for those that are not professionals in the field of quality analysis. For example, if the user wants to see only quality aspects of the data as it will be later used in a data mining application, the service will automatically select the adequate dimensions and the configuration file will be set up. (In the following sections we will go in detail for this possibility) Also, there is a possibility for the user to select an interval of values to inspect only.

4.2 Data Exploration tool as a support component of the quality service

The exploration of the output is the final step of a quality analysis. Either the user can disclose the need to perform a preprocessing task before making available the analyzed data object or it can emerge the need to run another quality analysis that is more focused on a specific aspect of the data object, or a new quality analysis with different settings over the quality dimensions.

Until now it was possible to capture the results from the quality analysis and explore them, but yet in a very inconvenient way for users that are not professionals in the field of data analysis. Also, even for professionals it can be hard to explore the results and draw conclusions because as we saw the analysis can be explored on three levels in the same time and that can cause a lot of confusion.

**4.2.1 Guided tool that will increase the level of knowledge for data analysis**

Having in mind that data quality in Big Data is quite a new thing in the literature we are aware that a lot of newcomers would will be brought in contact with quality services like the service offered by Bigsea. In the section before we saw how hard can be to navigate through the results and understand the real value of our data. To help the user in navigation, but also to learn more about the quality measures and levels of granularities, we thought of a guided exploration tool. We can go through a set of features that can take place in such a tool with the following examples:

* *While exploring global granularity it would be useful to show all dimensions according to a predefined scale. For example, if the score is bad mark the dimensions with red, otherwise if the score is good mark it green*
* *Let the user group the dimension by a processing task that will be done on the data set afterwards. For example:**A user just did the quality analysis manually, choosing all three level of detail and now it is interested to see if the data source would be appropriate for a processing task that requires only correct data. An exploration tool can save the user from another analysis over the data and show the quality dimensions which are the most important for such processing (in this case that would be Precision, Distinctness, Consistency, and Accuracy as dimensions).*
* *Always state the importance of exploring deeper granularity since it can bring more precise information*
* *While exploring attribute level of granularity let the user know which attribute is giving good scores according the dimension*
* *While exploring value level of granularity suggest the user to explore each record independently and see how it performed in all other dimensions*
* *Let the user know how important it is to custom filter the results*

**4.2.2 Retaining consumption and preprocessing tasks**

The idea of the Bigsea project is to propose an architecture that will be able to predict resource consumption of Big Data Analytics applications. Having this on mind we know that processing Big Data analysis can take a lot of time and energy and with that the cost of “value quality” will increase a lot. To save the user from iterative analysis until the fulfilled results are achieved, a simple data exploration tool can be practical. We will go through some scenarios that can illustrate how such a tool can be more convenient instead of a new processing task.

**Scenario 1: *Assume we want to find the bus lines that have the highest frequency and in the same time having completeness frequency greater than 0,6. In order to perform this we need to perform to separate tasks over the quality service, but instead if using an exploration tool we can filter the set of data and save only the records that are satisfying our condition.***

**Scenario 2: *A user wants to make statistical analysis over the data source and it is interested only in data records without duplication, and in the same time without missing values. In order to perform this, it is enough to do a custom filtering on which the user will ask to receive all records that have completeness missing set to 1 and distinctness set to 1.***

**Scenario 3: *Assuming we did the quality analysis and we just remember that we only need to know how the CODLINHA 101 performed in all the dimensions. Without making a new analysis this would take a lot of time, but with a tool could be calculated in seconds.***

**Scenario 4: *A user just did the analysis and while exploring the results finds out that the results of completeness frequency over attribute level are bad. To explore further and without making a new analysis, an exploration tool could group the records of each attribute into sets formed by a predefined threshold. Once exploring the sets with low values, the user can see how all the values with low frequency perform as a group comparing to all other attributes results.***

**Scenario 5: *Extra feature that could be handy to a user would be the possibility to group by name, value, and range, all the records over value level of granularity. Afterwards an export option into excel file could save a lot of time to search and insert each row separately.***

4.3 Working logic and complexity

For a data exploration tool to be able to process all the output data it should be able to manipulate the results in a fast manner. All the results saved by the quality service depending of the granularity are kept in JSON files. The size of the files goes from smaller to bigger as the granularity goes from higher(global) to lower(value) level. The only exception that can occur is the file that carries results about consistency as quality measure over value level of granularity. It can happen that the user requested to check for an association rule that as antecedent value takes more than two attributes together.

To be able to show the data on request by the user such a tool must read from the folders and present the results in a user-friendly manner. The tasks that require the less complexity are the ones where the user is interested to see results just as they are saved by the quality service. In particular, global, attribute, and value level of granularity results are kept in single files which means the data tool has to locate the right directory and read the file.   
Contrary, the tasks that require the highest complexity and processing are the ones that require reading results from multiple files in the same time. Such a case is when the user wants to perform a custom filtering over records of an attribute over a set of dimensions (read scenario 1 from point 4.2.2).

In contemplation of the working logic and complexity that such a tool would require a possible solution would be to set a server powerful enough to process all the tasks and pass the final results in a user interface screen. In the following points we will go through the idea of how the tool should be implemented.

4.4 Tool Implementation

In the following part we will go through the implementation of a data exploration tool by the ideas and discussions we present so far in the chapter. First, we will see approach in its creation, which technologies have been used to develop the tool, workflow, and feature exploration.

4.4.2 Approach in creating the exploration tool

As we mentioned before, the idea of the exploration tool is to provide support for all user, either professionals in data analysis, or beginners in the field.

Having in observance the application tool for setting the analysis that was also developed as part of the EU-BRA project within Politecnico Di Milano, this thesis is providing a similar mode of exploration just as the one used when setting up the quality analysis in first place.  
The idea to repeat the approach could prepare the user for a new quality analysis, and even more to get familiar with the concepts of quality analysis.

First point of interest for us is the identification of the Data Object of interest. That means a specific part of the data source (it could be the case when more than one data source is selected) should be analyzed. This is as important as the selection and configuration of quality dimensions because it allows the user to learn about separate portions of the data source. Having in mind this, the user can re-run the quality analysis many times and be confident in using the data in a further software.

Our second point of interest are three level of granularities on which the user can choose how to process the quality analysis: global, attribute, and value level of granularity. This is a continuance of the process of object identification.   
  
 A core part of a configuration of a quality analysis covers the selection of quality dimensions as measures of the value of the data itself. We gave detailed information in the previous chapter but for sake of clarity we should recap that there are two ways a user can select the quality dimensions, and they will be our third point of reference:

* Automatically - predefined configuration with the help of the application tool
* Manual – chosen by the user

**4.4.1 Technologies used**

The user interface of the project is developed in HTML/JavaScript, while the server side is based on Node.js.

JavaScript is a high-level, interpreted programming language. It is characterized as dynamic, prototype-based and multi-paradigm. Together with HTML and CSS provides interactive and reliable online programs. On the other side, Node.js represents an open-source, cross-platform JavaScript run-time environment for server-side scripting. The communication between the two parts is crucial since all the information is being first processed on server side and rendered on the front end.

Each page of the tool is dynamically built in such a manner that first instantiates an object of its type (for example Attribute granularity page instantiates attribute class) from which a HTTP AJAX request is sent towards the server for content retrieval. Each HTTP request from the front-end is identified on server-side by a uniform resource locator (URI). Once the request is processed, a response is being forwarded in a JSON format. Follows the list of resources used by the application:

* “/getSources” a URI used for retrieval of the data sources available
* “/results” a URI used for retrieval of the global granularity
* “/getAttributes” a URI used for retrieval of attributes of a dimension
* “/value” a URI used for retrieval of value granularity
* “/custom” a URI used for setting custom filtering

Note that all URIs require query attributes so that it will be feasible for the server to determine the source, dimension, or attribute of interest.

4.4.3 Navigation models in the UI

In this section we will go through the dynamic model of the UI. We will accomplish that by providing various diagrams to describe the interaction between objects and other components of the system. Later we will go through each page and feature in detail. Follows a set of navigations models from the user experience point of view.

4.4.3.1 Homepage

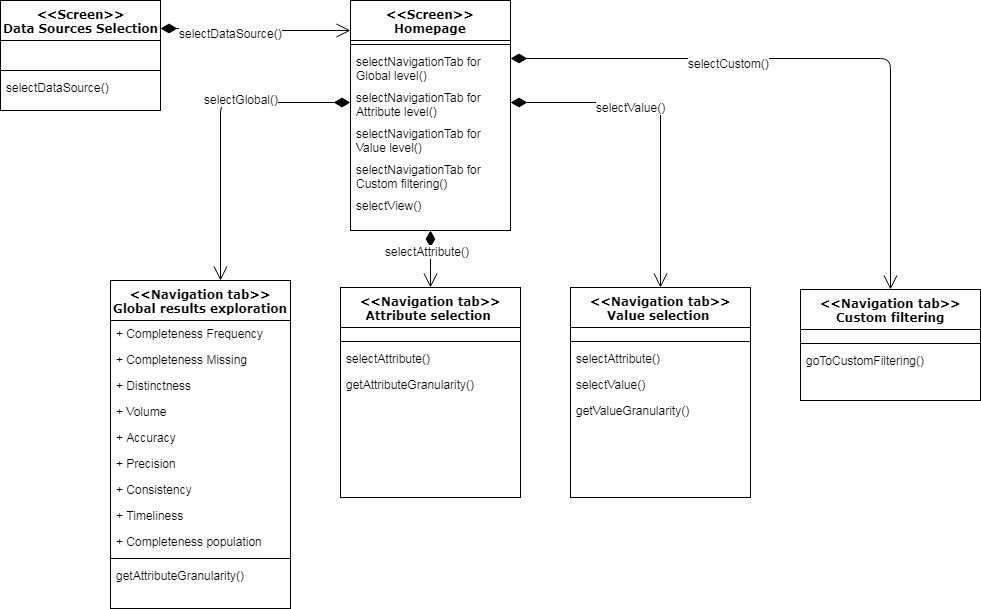


Figure 18: Homepage of the UI

4.4.3.2 Attribute page

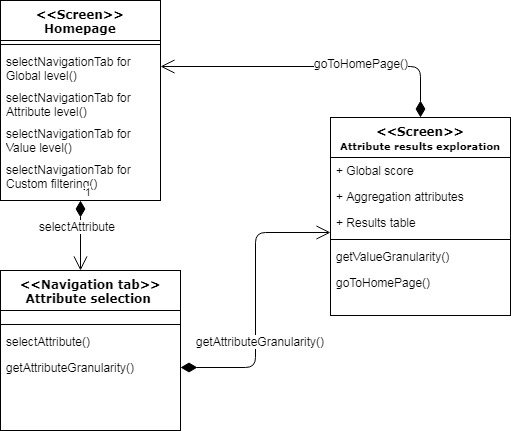


Figure 19: Exploration on attribute level

4.4.3.3 Value granularity and extra features

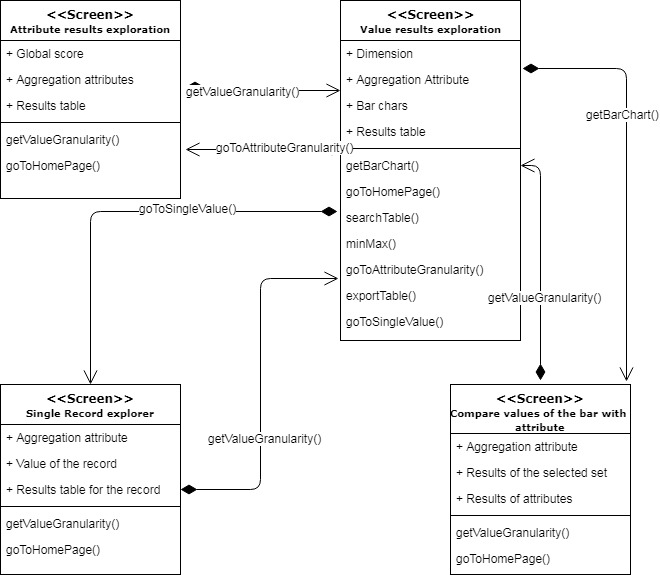


Figure 20: Exploration on value level

4.4.3.4 Custom filtering

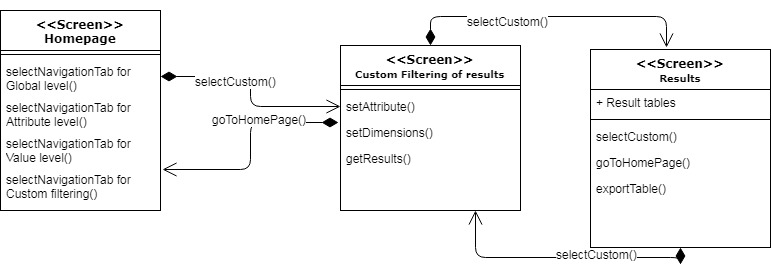


Figure 21: Custom filtering

**4.4.4.1 Data source selection**

When accessing the exploration tool, the user is asked to select the data source of interest.



Figure 22: Selection of a data source

4.3.2 Home page

After selecting the data source on which the user wants to explore the results, the UI makes a request on server side to get access to the results concerning the data source of selection. At first the UI shows to the user the results concerning global level of granularity. As it can be seen from the image below, all the quality dimensions are placed in equally sized boxes with the scores accordingly.

In order to help the user, see how good the scores by each quality dimension are, the UI classifies the results for each quality dimension by a set of predefined values that act as a threshold. For instance, if completeness frequency global scores are below 0,35 the UI marks them as bad results and scores above 0,8 as high results. In example, if the global score for completeness frequency in our analysis is 0,30 the UI will mark the box with red border. Otherwise, if it is above 0,8 the box will be marked with green color. All the values that are in between thresholds will be marked with yellow color.

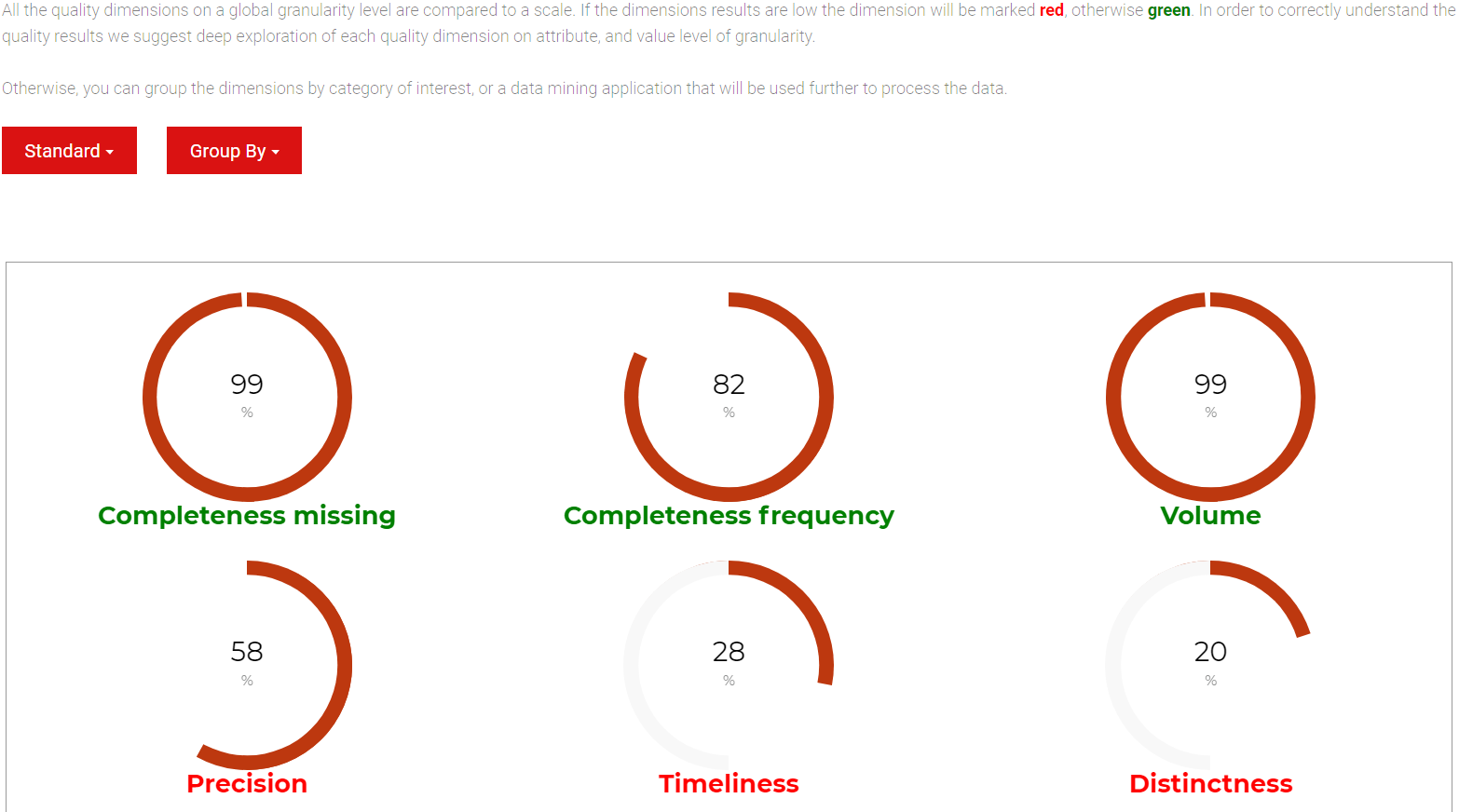


Figure 23: Home screen of the UI

For each quality analysis the selection of dimensions and their configuration can be different. As we said, the user itself chooses when setting the analysis which dimensions will be used to perform the quality analysis:

* Custom selection
* Group dimensions by category
* Group dimensions by data mining application
* Automatic selection

In the UI we want to implement this once more, since it will bring different view of the scores, and it can save time for setting a new analysis.

Assuming the user choose only automatic selection of dimensions. In this case all the application tool selected all available dimensions according to the previously defined object model and performed an analysis. Next when the analysis is done the UI will show the global results when the user enters in the home page exactly as they were exported by the application tool.  
The idea is that the UI will allow the users to manipulate this dimensions and group them by category or data mining application. In this way the user gets a standalone view that will show results as when doing a new analysis and grouping the dimensions by category or data mining application. In the image bellow we can see how the UI offers grouping of the quality results by a data mining application.

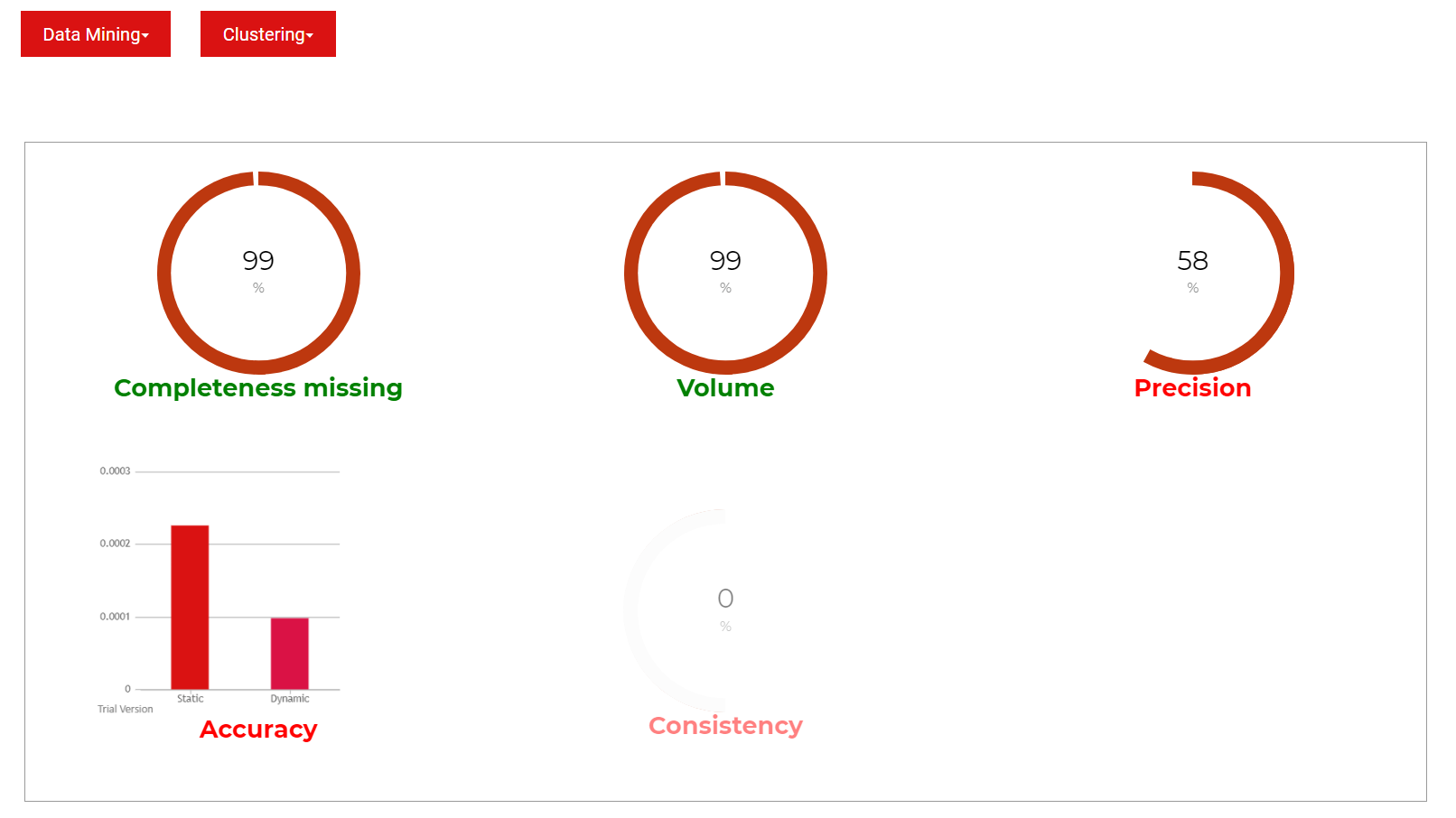


Figure 24: Data Mining view of dimensions

This feature is of a high importance because it can help an unexperienced user to determine which quality dimensions should be observed for the further usage of the data. In the case above the user intends to see how the data performed for dimensions that are associated to a data mining classification, since later the data will be used in such type of application for further processing. Beside of this configuration the user can select view within:

* Data Mining grouping where the application is Clustering
* Data Mining grouping where the application is Association Rule
* Category grouping where we associate Volume
* Category grouping where we associate Time Validity (Timeliness as dimension)
* Category grouping where we associate Correctness (Precision, Distinctness, Consistency, Accuracy as dimensions)
* Category grouping where we associate Completeness (Completeness Missing, Completeness Frequency, Completeness Population)

Of course, the user can always switch back to the first presentation of the quality dimensions. Note that if the quality analysis was performed with custom choice (selecting just some of the dimensions) it can happen that in the views some dimensions will be missing. To escape a confusion, the UI will warn of such a case.

4.3.3 Attribute level of granularity

Usually, when an unexperienced user uses the application selects the whole data source as an object of interest. Leaving the whole data source as a data object of interest can give a wrong information for the quality of the data. There can be parts of the source that are inadequate and are giving wrongful data which can create confusing results in a further usage.

The second level of depth regards the selection of attributes from the data source. This means that the user is interested in specific attributes of the source and wants to inspect their quality. This can give better quality results since each selected dimension is being run over each selected attribute. Think of an example where we performed only global analysis over a source and we get low scores for completeness missing. There is no way in this mode of inspection to know from which part of the data source the missing rows are coming. In fact, when selecting attributes of interest, we can get this information correctly.

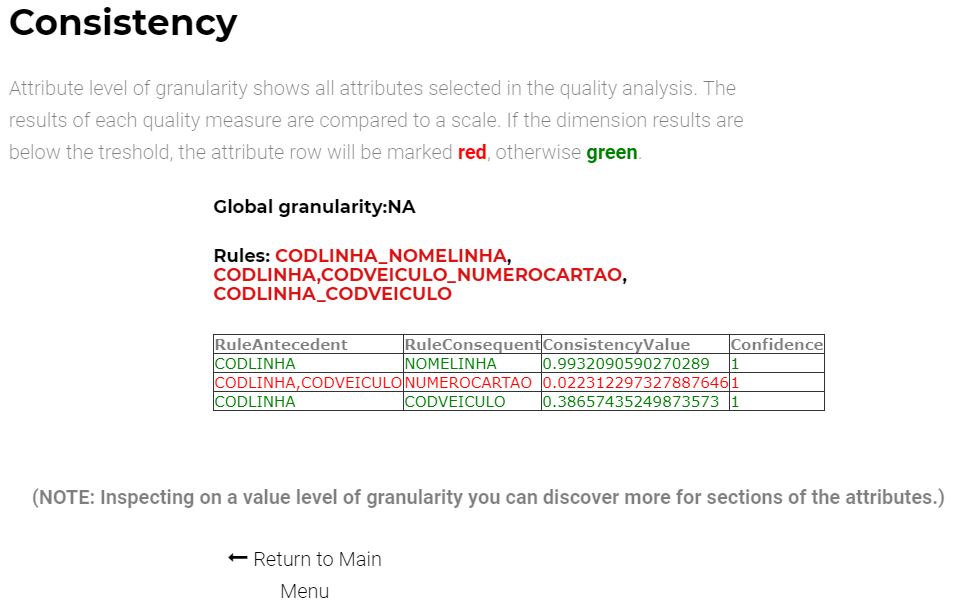


Figure 25: Attribute level of granularity

In the image above, we can see a screen of the attribute level of granularity of the UI. At the beginning section of the page the user is guided through the logic of marking attribute scores. The classification remains the same as on global depth, all attributes scores are compared to a threshold depending on the quality dimension of interest. This can help the user when setting a further analysis to include/exclude a specific attribute. Further, the user is advised to explore results on deeper level as they can give more details of the selected attribute.

4.3.4 Value level of granularity

Further, and last level of depth regarding the selection of the data object is value granularity. When setting up the analysis in the application tool users were also allowed to choose values for each attribute included in the data object. There are two types of possible selections for each attribute: set of values selections and an interval of value selection. This can have huge impact on the quality of the results because we can know which exact values are having poor quality results, and so can be omitted from further inspection, or in opposite, they can be the core of a new quality analysis.   
The exploration tool is giving a big accent of the importance of the value level of granularity and presents to the user interval and detailed exploration even if it was not stated in the quality application tool when configuring the analysis.

For each quality dimension the user is exploring at the time (inside of an attribute) the UI offers further view on a value level of granularity. In the bottom part of the page we can see values of the attribute of the attribute of interest with their quality scores accordingly placed in a table.

Further, the user can manipulate the data in the table by his/her choice. There is a filtering section that goes through the scores of the results. Also, it is possible to make a search over all rows of the table. Of course, after filtering it is possible to export the data in xls format, which can find further usage to the user.

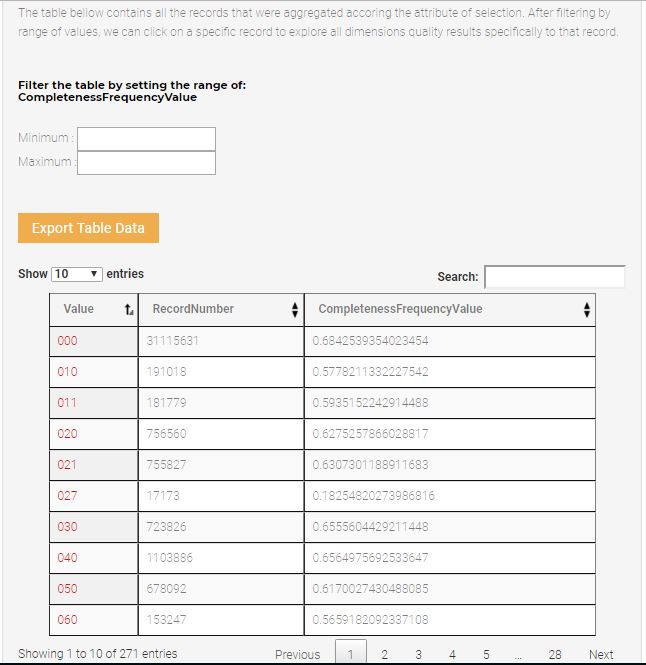


Figure 26: Value level of granularity (table filtering)

With the feature of filtering the user can manipulate the data and, in that way, learn how most of the values are performing by the selected dimension. Small dimensions scores for a value can help us ignore the value in the next testing, and in opposite, high scores for a value can lead us to a new analysis where we can state exact value for an attribute.

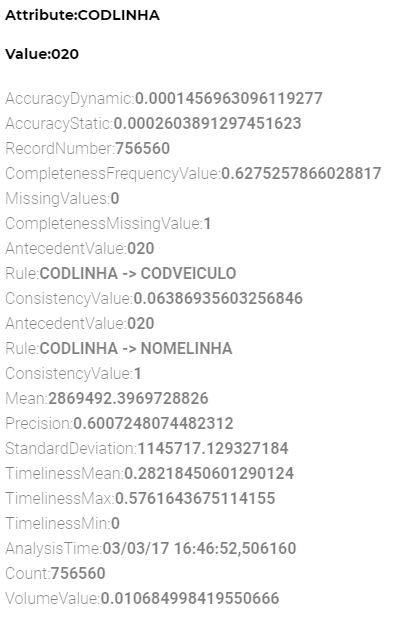
Even before starting with a new analysis, the user can click on a value and explore more information concerning it. This leads us to the next feature of the UI.  
  


Figure 27: exploring a single value of an attribute

For the sake of clarity, this feature will be explained through the following scenario:

*Assuming the data source on which the analysis is done is the one regarding the BusUsers from the city of Curitiba, Brazil. The data in the source is collected from the ticket validation procedure in each of the buses of the city. The log is composed of the following attributes:*

* *CODLINHA – the code assigned to the bus line in which the ticket validation has been performed*
* *NOMELINHA – name of the bus line*
* *CODVEICULO – code of the vehicle*
* *NUMEROCARTA – code of the ticket card*
* *DATAUTILIZACAO – timestamp of the validation*

*Assuming the user explores value level of granularity for CODLINHA as an attribute and Volume as a quality dimension. Volume as a quality dimension will report which are the values that occur with low or high quality. So, when the user finds which code line is the most frequent one it could be useful to know more about that line. For instance, he could run a new quality analysis where the CODLINHA will have the value of the most frequent one. In order to save the user from doing that, the exploration tool offers a view where we can get to know how the bus line performed on value level of granularity for all of the available dimensions.*

The page that inspects value level of granularity offers another feature that brings in a way a deeper inspection and classification of the whole set of values that one attribute can have.   
The idea is that for each quality dimension that the user is exploring on value level of granularity the application offers grouping of the set of values.

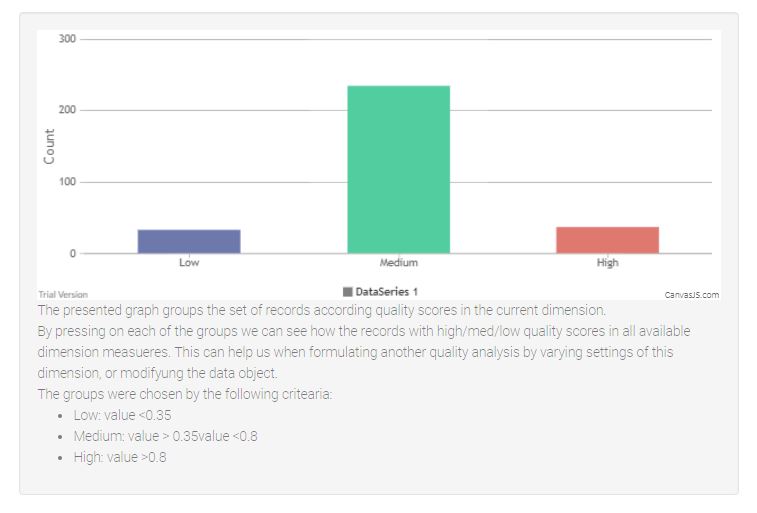


Figure 28: Values classification

The classification is done according a predefined quality values that act as reference threshold for low/high scores.

*For example, assuming the user is interested in value level of granularity scores by completeness frequency. The first threshold is set at 0.35. This means that every value of the attribute of interest that has a score less than the low threshold will be placed in the set of low values. In opposite all the values that have scores above 0.80 threshold are placed in the set of high scores.* *All the values in between the thresholds will be placed in the medium set.*

Of course, the presentation of the graph provides a better understanding of the scores, but even a better exploration can be done of each of the sets. By clicking on a set from the graph, the application offers another view.

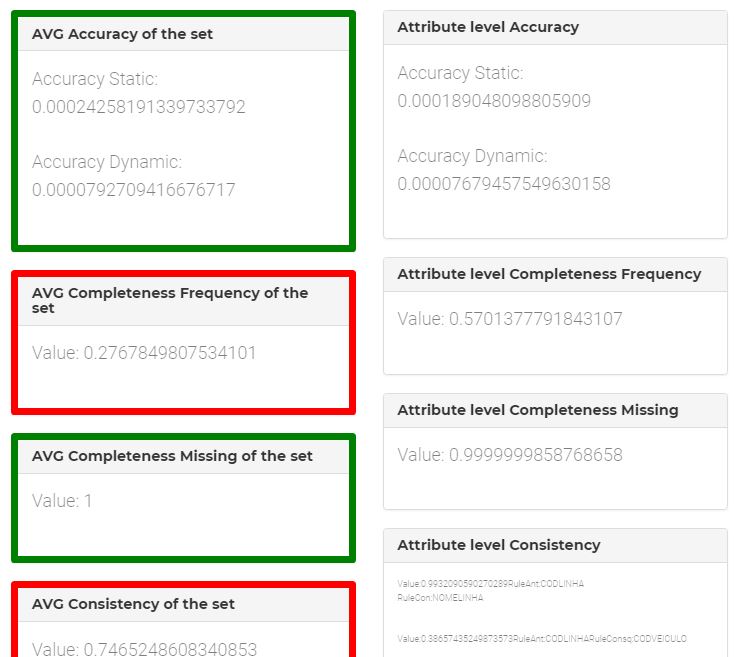


Figure 29: Comparison between values of the selected set and attribute level

All the values of the set are forming an average for each quality dimension used in the quality analysis. This average is compared to the quality dimension scores of attribute level of granularity (\*Note that this is the same attribute on which we inspect value level of granularity). This feature can be of a big interest for a professional in the field of quality analysis since it offers a comparison that couldn’t be done when setting up an analysis.

4.3.4 Custom filtering

On the home screen the UI is offering an alternative exploration and filtering of the data source. When choosing this option, the UI shows another view where the user can see all available dimensions and filter them by choice. The filtering is performed on value level of granularity, where the user chooses the attribute of interest.

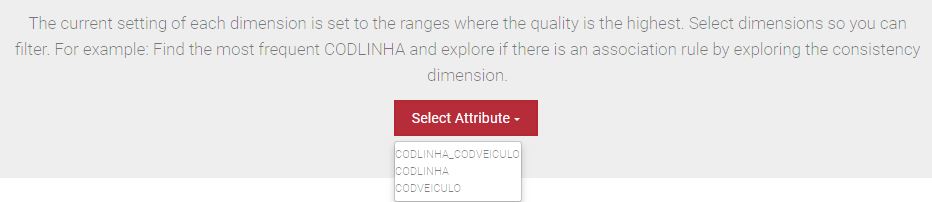


Figure 30: selection of attribute in custom filtering

All dimensions are filtered by choosing a range of values. The default setting for each quality dimensions is on a predefined threshold for which it is assumed that the scores are in a high range (marked with green).

The idea is to combine two or more quality measures over values of a same attribute. For instance, the user would like to know the most frequent bus lines (CODLINHA) and see if there is an association rule associated to them. To perform this, we must first select the ranges of volume and consistency dimensions in the high ranges and exclude all the other dimensions when performing the query.

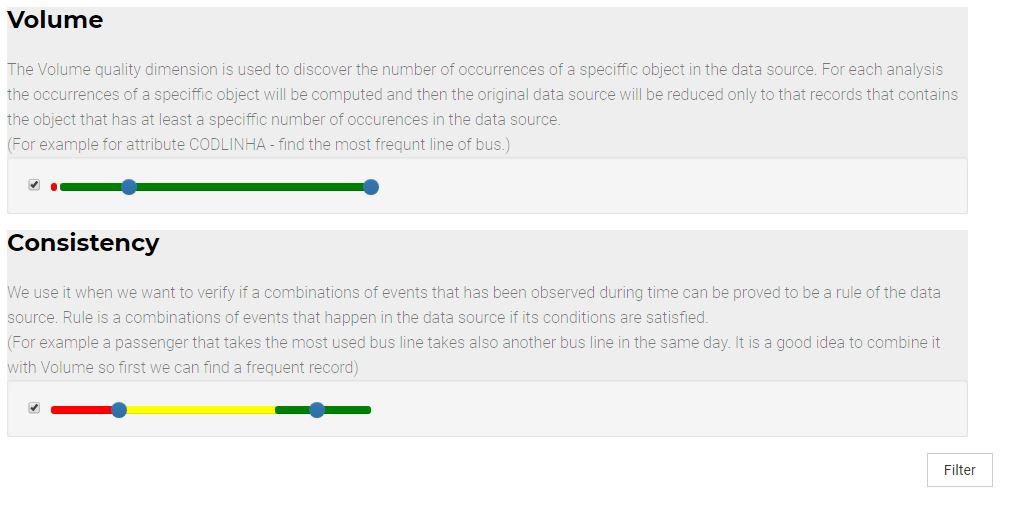


Figure 31: filtering over two dimensions by choice

In the result page the user can see tables related to volume and consistency dimensions with values that match the requirements. Most importantly, there is a merged table that shows values in common (in this case all the CODLINHA lines that are in common for the selected dimensions).

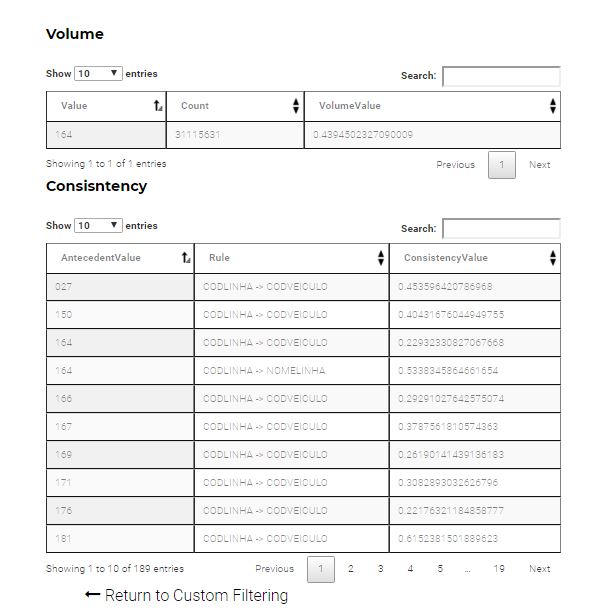


Figure 32: results of custom filtering feature

The settings of a custom filtering could be done on all attributes and with a set of dimensions of our choice. That brings a lot of combinations and possibilities to know more about the data. Having the values in common between the dimension can help the user to define precise analysis.

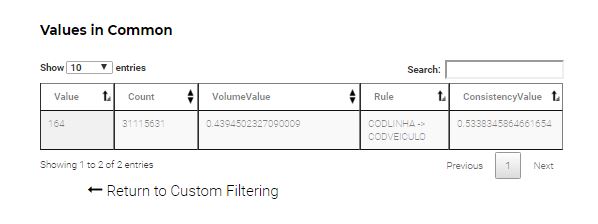


Figure 33: Values satisfying filtering criteria’s

# chapter 5

**Conclusion**