Universidad
del Cauca®

uc3m | Universidad Carlos III de Madrid

TESIS DOCTORAL

En convenio específico de cotutela: 23–32. 7 – 088 2015

UNIVERSIDAD DEL CAUCA
Programa de Doctorado en Ingeniería Telemática

UNIVERSIDAD CARLOS III DE MADRID
Programa de Doctorado en Ciencia y Tecnología Informática

Framework for Data Quality in Knowledge Discovery Tasks

AUTOR
David Camilo Corrales Muñoz

DIRECTOR
Dr. Juan Carlos Corrales
CO-DIRECTOR
Dr. Agapito Ledezma Espino

Noviembre 2018

TESIS DOCTORAL

FRAMEWORK FOR DATA QUALITY IN KNOWLEDGE DISCOVERY TASKS

Autor: David Camilo Corrales Muñoz

Director: Dr. D. Juan Carlos Corrales
Co-director: Dr. D. Agapito Ledezma

Tribunal Calificador		Firma
Presidente:	Fernando Fernández Rebollo	....
Vocal:	Juan Pedro Valente	....
Secretario:	Gustavo Adolfo Ramírez	....

Calificación: ....

Fecha: ....

Agradecimientos

Cuando te sientas a escribir estas palabras y te colocas a pensar en cada una de las personas que te apoyaron a lo largo de tu doctorado, te das cuenta que estas líneas no son suficientes para agradecerles toda la contribución que de alguna u otra forma han realizado. Sin embargo, no está demás decirles, infinitas gracias.

En especial a mis padres Fredy y Stella, por sus sabios consejos, por infundirme desde muy pequeño el significado de la responsabilidad, forjar mi carácter y enseñarme a levantar, con más fuerza y más ganas cuando las cosas no salen bien.
A mi hermano Juan Carlos, que ha sido siempre mi modelo a seguir. La persona que más admiro y la que me ha regalado infinidad de consejos como director de tesis y como hermano.

A Mariale, por ser mi inspiración, por contagiarme de su optimismo, por apoyar mis decisiones, por corregirme cuando estoy equivocado y por soportarme en momentos difíciles.

A mis directores, el Profe Agapito y Juan Carlos, por guiarme a lo largo de este camino, por brindarme sus sabios consejos, y por la formación adquirida como investigador.

A los chicos de Grupo CAOS, en especial a Mari Paz, Jose Antonio, Óscar, Germán y la Profe Araceli que me hicieron sentir en casa durante mi estadía en España, por todos los cafés y comidas que compartimos juntos, por abrirme las puertas de sus casas.

A los chicos del Grupo GIT, especialmente a mis amigos Julián, Ángela, Emmanuel e Iván, y a los Ingenieros Álvaro y Gustavo por el apoyo en la formación como Doctor en Ingeniería Telemática.

Resumen

Actualmente la explosión de datos es tendencia en el universo digital debido a los avances en las tecnologías de la información. En este sentido, el descubrimiento de conocimiento y la minería de datos han ganado mayor importancia debido a la gran cantidad de datos disponibles. Para un exitoso proceso de descubrimiento de conocimiento, es necesario preparar los datos. Expertos afirman que la fase de preprocesamiento de datos toma entre un 50% a 70% del tiempo de un proceso de descubrimiento de conocimiento.

Herramientas software basadas en populares metodologías para el descubrimiento de conocimiento ofrecen algoritmos para el preprocesamiento de los datos. Según el cuadrante mágico de Gartner de 2018 para ciencia de datos y plataformas de aprendizaje automático, KNIME, RapidMiner, SAS, Alteryx, y H20.ai son las mejores herramientas para el desucrimiento del conocimiento. Estas herramientas proporcionan diversas técnicas que facilitan la evaluación del conjunto de datos, sin embargo carecen de un proceso orientado al usuario que permita abordar los problemas en la calidad de datos. Además, la selección de las técnicas adecuadas para la limpieza de datos es un problema para usuarios inexpertos, ya que estos no tienen claro cuales son los métodos más confiables.

De esta forma, la presente tesis doctoral se enfoca en abordar los problemas antes mencionados mediante: (i) Un marco conceptual que ofrezca un proceso guiado para abordar los problemas de calidad en los datos en tareas de descubrimiento de conocimiento, (ii) un sistema de razonamiento basado en casos que recomiende los algoritmos adecuados para la limpieza de datos y (iii) una ontología que representa el conocimiento de los problemas de calidad en los datos y los algoritmos de limpieza de datos. Adicionalmente, esta ontología contribuye en la representacion formal de los casos y en la fase de adaptación, del sistema de razonamiento basado en casos.

Abstract

The creation and consumption of data continue to grow by leaps and bounds. Due to advances in Information and Communication Technologies (ICT), today the data explosion in the digital universe is a new trend. The Knowledge Discovery in Databases (KDD) gain importance due the abundance of data. For a successful process of knowledge discovery is necessary to make a data treatment. The experts affirm that preprocessing phase take the 50% to 70% of the total time of knowledge discovery process.

Software tools based on Knowledge Discovery Methodologies offers algorithms for data preprocessing. According to Gartner 2018 Magic Quadrant for Data Science and Machine Learning Platforms, KNIME, RapidMiner, SAS, Alteryx and H20.ai are the leader tools for knowledge discovery. These software tools provide different techniques and they facilitate the evaluation of data analysis, however, these software tools lack any kind of guidance as to which techniques can or should be used in which contexts. Consequently, the use of suitable data cleaning techniques is a headache for inexpert users. They have no idea which methods can be confidently used and often resort to trial and error.

This thesis presents three contributions to address the mentioned problems: (i) A conceptual framework to provide the user a guidance to address data quality issues in knowledge discovery tasks, (ii) a Case-based reasoning system to recommend the suitable algorithms for data cleaning, and (iii) an Ontology that represent the knowledge in data quality issues and data cleaning methods. Also, this ontology supports the case-based reasoning system for case representation and reuse phase.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Context

Due to advances in Information and Communication Technologies (ICT), today the data explosion in the digital universe is a new trend [1, 2, 3]. The vast amount of data coming from different sources such as social networks, messenger applications for smart-phones, IoT, etc. The Forbes magazine reports an increase of data every second for every person in the world of 1.7 Megabytes from 2020 [4].

To maintain a competitive edge, organizations need to take advantage of the large amount of data to extract useful knowledge for making feasible decisions [5, 6]. These benefits facilitate the growth of organizational locations, strategies and customers. Decision makers can utilize the more readily available data to maximize customer satisfaction and profits, and predict potential opportunities and risks. To achieve it, the data quality must be guaranteed. Data quality is directly related to the perceived or established purposes of the data. High-quality data meets expectations to a greater extent than low-quality data [7].

Nevertheless, the majority of organizations are pervaded with poor quality data [8, 9]. The appearance of such poor quality data and the presence of various errors significantly reduce the usability and creditability of the information systems and can have a moral and financial impact on the members of the organization and its associated stakeholders. A survey conducted in [10] revealed that data quality problems cost US businesses 611 billion dollar a year.

This thesis address the data quality issues in knowledge discovery (KD) tasks (classification and regression) through a conceptual framework to provide the user a guidance to address data problems, case-based reasoning system to recommend the suitable algorithms for data cleaning and an ontology that represent the knowledge in data cleaning.

1.2 Motivation

For a successful process of knowledge discovery (KD), there are methodologies such as the Cross Industry Standard Process for Data Mining (CRISP-DM) and Sample, Explore, Modify, Model and Assess (SEMMA). CRISP-DM contains two steps for data treatment: Verify Data Quality and Clean Data, while SEMMA the Mo-dify phase. Although the knowledge discovery methodologies define steps for data treatment, these not tackle the issues in data quality clearly, leaving out relevant activities [6, 11].

In this sense, a poor data preprocessing phase can potentially impact on the remainder of the phases in the knowledge discovery process. Data preprocessing is an essential step in knowledge discovery projects [12, 13]. It deals with preparing data to be stored, processed or analyzed and with cleaning it from unnecessary and problematic artifacts. It has been stated that preprocessing takes 50% to 70% of the total time of knowledge discovery projects [12, 13]. Data cleaning tasks are at once the most tedious and the most critical task. Failure to provide high data quality in the preprocessing stage will significantly reduce the performance of any data mining project. Hence, the phrase "garbage in garbage out" becomes true in the case of a data mining project [8]. In the following, we highlight the most relevant preprocessing challenges (data quality issues):

- **Missing values**: refers to when one variable or attribute does not contain any value.
- **Outliers**: these are observations which deviate much from other observations and are suspicions that it was generated by a different mechanism.
- **High dimensionality**: is referred as when dataset contains a large number of features.
- **Imbalanced class**: is considered when a dataset exhibits an unequal distribution between its classes.
- **Mislabelled classes**: instances that are contradictory (duplicate samples have different class labels).
- **Duplicate instances**: represent instances with same values.

In this thesis we address the data quality problems mentioned above.

1.3 Problem statement

The methodologies of knowledge discovery mentioned above define a data processing phase. In CRISP-DM it is called Data Preparation phase, while in SEMMA is named Modify stage. Nevertheless, these methodologies do not explain how to find and what to do when data quality issues are present in data processing phase.

Several knowledge discovery tools simplify the analysis and management of data. According to Gartner 2018 Magic Quadrant for Data Science and Machine Learning Platforms [14], KNIME [15], RapidMiner [16], SAS [17], Alteryx [18] and H20.ai [19] are the leader tools for knowledge discovery. These KD tools provide different techniques and they facilitate the gathering, application, inspection, and evaluation of data analysis and their results, however, these KD tools lack any kind of guidance as to which techniques can or should be used in which contexts [20]. Consequently, the use of suitable data analysis technique is a headache for inexpert users. They have no idea which methods can be confidently used and often resort to trial and error [20].

Thus, in this thesis were addressed the problems identified above.

1.4 Research Questions

Based on the considerations previously described, this doctoral thesis rises the research questions:

1. *How to assess the data quality in knowledge discovery tasks*?

2. *How to select the right data cleaning algorithm for solving a data quality issue*?

1.5 Research Purpose and Objectives

The purpose of this research is to develop a framework for analysis of data quality issues in knowledge discovery tasks through artificial intelligence based techniques. The research purpose was achieved through:

1. Define a conceptual framework to guide to user in data quality issues in knowledge discovery tasks (classification and regression).

2. Establish strategies that advise the suitable data cleaning algorithm to user for solving the data quality issue.

3. Build a mechanism that gathers data cleaning algorithms to solve the data quality issues identified by the framework.

4. Develop and evaluate experimentally a prototype that tests the mechanisms and strategies of the framework for data quality in knowledge discovery tasks.

1.6 Contributions

This section lists the main contributions of the PhD thesis. Each contribution is aligned with the objectives 1, 2 and 3 mentioned above. The objective 4 gathers the results of the first three objectives.

- A conceptual framework to provide the user a guidance to address data quality issues in knowledge discovery tasks. To construct the conceptual framework we adapted the metodology "*Building a Conceptual Framework: Philosophy, Definitions, and Procedure*" [21].

- An ontology that gathers the data cleaning algorithms to solve the data quality issues. This ontology allow to know the suitable data cleaning approach with respect to data quality problem.

- A Case-base reasoning to advise the suitable algorithm for data cleaning in classification and regression tasks.

Figure 1.1 shows the relations among contributions explained above. The conceptual framework guides the process to address data quality issues. The case-based reasoning system supports to the conceptual framework in each component, recommending the suitable data cleaning algorithm based on past experiences. Data cleaning ontology represent the knowledge of data quality issues and data cleaning algorithms for classification and regression tasks. This ontology lists methods of a data cleaning approach. Besides, this supports the CBR system in the case representation and reuse phase.

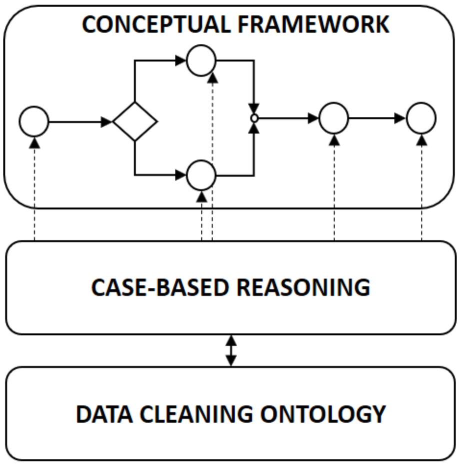


Figure 1.1: Contributions of PhD thesis

1.7 Outline

To ensure a comprehensive coverage of the research problems, existing methods and the proposed goals, this thesis is organized as follows:

- **Chapter 2** provides a comprehensive coverage of related works of data quality frameworks, data cleaning ontologies and case-based reasoning for data cleaning.
- **Chapter 3** details the conceptual framework to guide to user in the analysis of data quality issues.
- **Chapter 4** presents the case–based reasoning system to advise the suitable algorithm for data cleaning.
- **Chapter 5** describes the ontology for data cleaning in classification and regression tasks.
- **Chapter 6** shows the conclusions and future works.

1.8 Publications

This section lists the papers built from PhD thesis:

1.8.1 Accepted papers

- **Corrales, D. C.**, Ledezma, A., & Corrales, J. C. (2018) "From theory to practice: a data quality framework for classification tasks", *Symmetry*, 10(7), (JCR: Q_2). Parts of this work have been incorporated in this thesis, in Chapters 3 and 4.

- **Corrales, D. C.**, Ledezma, A., & Corrales, J. C. (2018). "How to Address the Data Quality Issues in Regression Models: A Guided Process for Data Cleaning", *Symmetry*, 10(4), (JCR: Q_2). Parts of this work have been incorporated in this thesis, in Chapter 3.

- **Corrales, D. C.**, Lasso, E., Ledezma, A., & Corrales, J. C. (2018). "Feature selection for classification tasks: Expert knowledge or traditional methods?". *Journal of Intelligent & Fuzzy Systems*. In Press (JCR: Q_3). Parts of this work have been incorporated in this thesis, in Chapter 3.

- **Corrales, D. C.**, Ledezma, A., & Corrales, J. C. (2016) ."A systematic review of data quality issues in knowledge discovery tasks". *Revista Ingenierías Universidad de Medellín*, 15(28), 125-150.

- **Corrales, D. C.**, Ledezma, A., & Corrales, J. C. (2015). "A conceptual framework for data quality in knowledge discovery tasks (FDQ-KDT): A Proposal". *Journal of Computers*, 10(6), 396-405 (SJR: Q_3).

1.8.2 Other published papers

- **Corrales, D. C.**, Lasso, E., Figueroa, A., Ledezma, A., & Corrales, J. C. (2018). "Estimation of coffee rust infection and growth through two-level classifier ensembles based on expert knowledge". *International Journal of Business Intelligence and Data Mining (IJBIDM)*, 13(4), 369-387.

- Castillo, E., **Corrales, D. C.**, Lasso, E., Ledezma, A., & Corrales, J. C. (2017). "Water quality detection based on a data mining process on the California estuary". *International Journal of Business Intelligence and Data Mining*, 12(4), 406-424. Parts of this work have been incorporated in this thesis, in Chapter 3.

- **Corrales, D. C.**, Gutierrez, G., Rodriguez, J. P., Ledezma, A., & Corrales, J. C. (2017). "Lack of Data: Is It Enough Estimating the Coffee Rust with Meteorological Time Series?". In *International Conference on Computational Science and Its Applications* (pp. 3-16). Springer, Cham.

- **Corrales, D. C.**, Corrales, J. C., Sanchis, A., & Ledezma, A. (2016). "Sequential classifiers for network intrusion detection based on data selection process". In *Systems, Man, and Cybernetics (SMC) IEEE International Conference* (pp. 001827-001832). IEEE Xplore.
- Castillo, E., **Corrales, D. C.**, Lasso, E., Ledezma, A., & Corrales, J. C. (2016). "Data Processing for a Water Quality Detection System on Colombian Rio Piedras Basin". In *International Conference on Computational Science and Its Applications* (pp. 665-683). Springer, Cham. Parts of this work have been incorporated in this thesis, in Chapter 3.
- **Corrales, D. C.**, Figueroa, A., Ledezma, A., & Corrales, J. C. (2015). "An empirical multi-classifier for coffee rust detection in colombian crops". In *International Conference on Computational Science and Its Applications* (pp. 60-74). Springer, Cham.

2. State of Art

This chapter presents the background related to the main topics addressed in this doctoral thesis. First, we explain the definitions: Data Quality Framework, Ontology and Case-Based Reasoning. Subsequently, we present the current literature that discusses Data Quality Frameworks, Data cleaning Ontologies and Case-Based Reasoning Systems, particularly that which focuses on knowledge discovery tasks. For each topic of the literature review (Data Quality Frameworks, Data cleaning ontologies and Case-Based Reasoning Systems), we show the shortco mings of the related works, and we mention our contributions.

2.1 Background

In this section, we presented four definitions, the first, methodologies for knowledge discovery and the remain definitions are related to the main contributions: data quality framework, ontology, and case-based reasoning.

2.1.1 Methodologies for Knowledge Discovery

In this subsection, we describe the methodologies for knowledge discovery (KD) from data, which are the most frequently used in machine learning and data mining projects. Considering that data quality is the core of the PhD thesis, we highlight the phases of KD methodologies that involve an analysis of data quality.

2.1.1.1 Knowledge Discovery in Databases (KDD)

The authors in [22] defined the Knowledge Discovery in Databases (KDD) as “the process to find valid, novel, useful and understandable patterns in data, to describe/predict the future behavior of some event”. Thus, the KDD process considers five phases (Figure 2.1):

- Selection: this stage refers to selection and creation a data set on which discovery will be performed.

- Preprocessing: in this stage, data are cleaned. It includes handling missing values, removal of noise and outliers detection.

- Transformation: this stage finds useful features to represent the dataset focused in the knowledge discovery task. The aim of this stage is to reduce the high dimensionality.

- Data Mining: in this stage, knowledge discovery tasks (classification, regression and clustering) are applied for pattern extraction.

- Interpretation/Evaluation: this stage involves the analysis of the extracted patterns and generated models from knowledge discovery tasks. In addition, the models are evaluated.

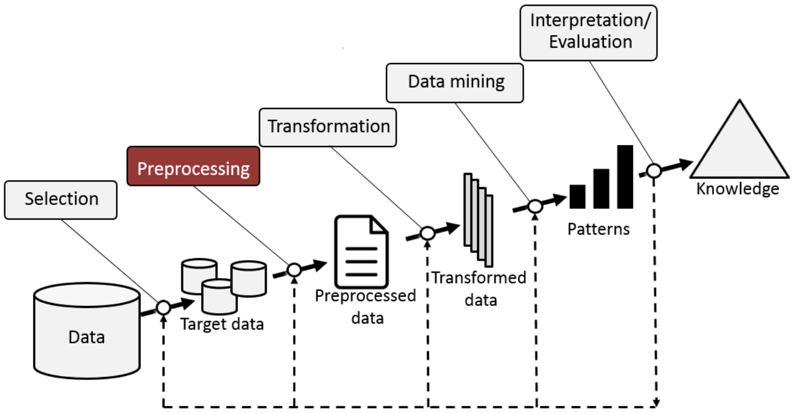


Figure 2.1: Phases of KDD process.

2.1.1.2 Cross Industry Standard Process for Data Mining (CRISP-DM)

CRISP-DM is a methodology for data mining projects [13]. The life cycle of CRISP-DM consists of six phases (Figure 2.2):

- Business understanding: this phase focuses on understanding of the domain from data mining problem perspective.

- Data understanding: in this phase the data are collected and analyzed (identification of data quality issues).

- Data preparation: this phase covers the activities related to construct the final dataset for the application of a knowledge discovery task.

- Modeling: depending of the knowledge discovery task selected, modeling techniques are used.
- Evaluation: in this phase, the models built in the Modeling phase are evaluated through performance measures.
- Deployment: this phase involves the deployment of the models in the real world.

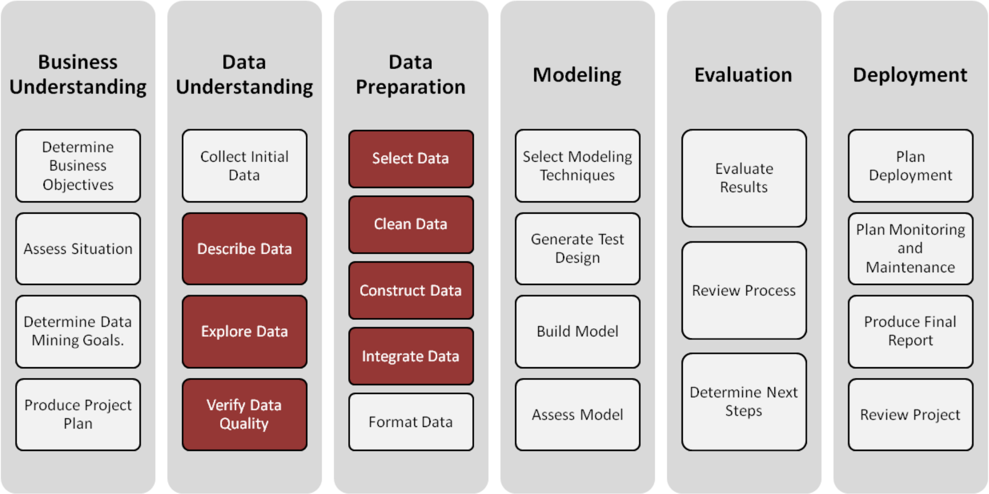


Figure 2.2: Phases and generic tasks of CRISP-DM.

Each phase contains a set of tasks as show Figure 2.2. The generic tasks highlighted in red color, involve activities of data quality. For example, the tasks of “Data Understanding” phase, examine and visualize the data quality. In case of “Data Preparation” tasks, correspond to data preprocessing.

2.1.1.3 Sample, Explore, Modify, Model and Assess (SEMMA)

In addition to the CRISP-DM, the SAS Institute developed a methodology called SEMMA [23]. The acronym SEMMA stands for Sample, Explore, Modify, Model, Assess (Figure 2.3):

- This methodology begins with a statistically representative sample of the dataset (Sample).
- Subsequently, SEMMA applies exploratory statistical and visualization techniques (Explore).

- After, the methodology suggests the selection and transformation of the most significant predictive variables (Modify).
- Next, the model is built based on variables to predict outcomes (Model).
- Finally, the model accuracy is evaluated (Assess).

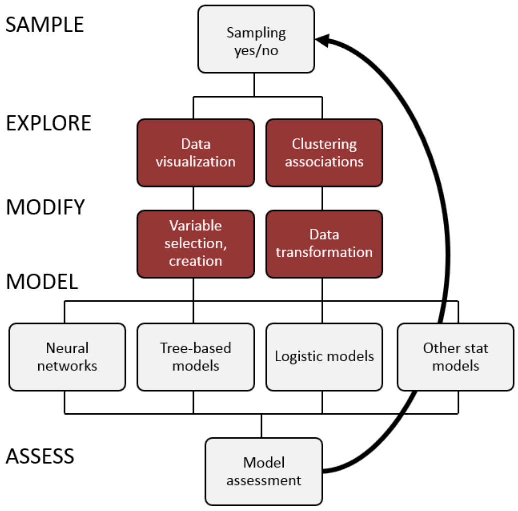


Figure 2.3: Stages of SEMMA process.

The stages highlighted in red color in Figure 2.3, correspond to activities related with data quality. Thus, the "Explore" stage, consists on the exploration of the dataset properties, and "Modify" stage is referred to modification of the data to focus the model selection process [23].

2.1.1.4 Data Science

Data Science refers to generalizable extraction of knowledge from data [24]. This process is focused in the representation, analysis of data, and relations among variables [25]. The data science process is composed with the next steps (Figure 2.4):

- The first stage involves the recollection of raw data (In Figure 2.4 Raw data is collected).
- Subsequently, the data scientist transforms the raw data into a format readable for a data analysis tool (In Figure 2.4 Data is processed).

- The “clean data” phase comprises the detection and cleansing of data quality issues as outliers and missing values
- The “exploratory data analysis” phase refers to analyzing the dataset through statistical and visual methods to summarize the main dataset characteristics.
- Machine Learning Algorithms and Statistical Models are selected for a specific knowledge discovery task in “Models and algorithms” phase
- In “Construction of reports” phase, the data scientist build reports of the raw and cleaning data, also of the algorithms and models used in previous phases.
- The last phase consist in build data product. This phase involves the deployment in the real world (In Figure 2.4 Build data product).

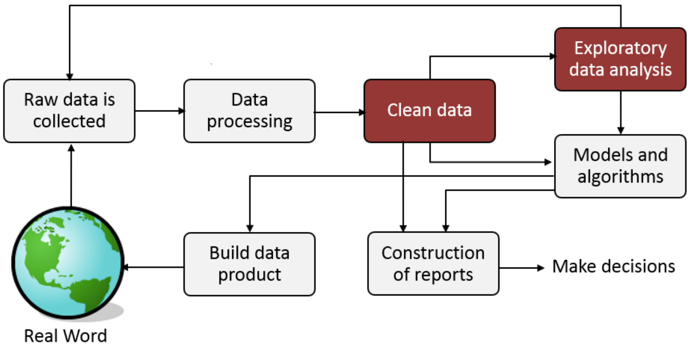


Figure 2.4: Data science process. Source: [24]

The analysis of data quality is made in the phases: “Data processing”, “Clean data”, and “Exploratory data analysis” (in Figure 2.4 the steps highlighted in red color).

2.1.2 Data Quality Framework

Data are representations of the perception of the real world. They can be considered the basis of information and digital knowledge [26]. Data quality is a critical factor to maintain consumers’ needs. The quality of data is defined by two related factors: how well it meets the expectations of data consumers [27] and how well it represents the objects, events, and concepts it is created to represent. In order to measure whether data meets expectations or is “fit for use” expectations and uses need to be defined [7].

For ensuring data quality in data management systems, we need to consider two relative aspects in evaluating the quality of data: the actual quality that can be evaluated at the data source and the expected quality that is required by the users at the users' views [28]. Authors in [29] affirm that the cost of poorly structured data produced in large amounts is solved by frameworks for the assessment of the data quality.

The Data Quality Frameworks seek to assess areas where poor quality processes or inefficiencies may reduce the profitability of an organization [30]. At its most basic, a data quality framework is a tool for the assessment of data quality within an organization [31]. The framework can go beyond the individual elements of data quality assessment, becoming integrated within the processes of the organization. Eppler and Wittig [32] add that a framework should not only evaluate, but also provide a scheme to analyze and solve data quality problems by proactive management.

A real case is presented by Deloitte Belgium [33]. This enterprise developed a Data Quality Framework to assess the data risks and data health. The framework analysis and provides insights into the root causes of poor data quality. Figure 2.5 presents the Data Quality Framework developed by Deloitte Belgium.

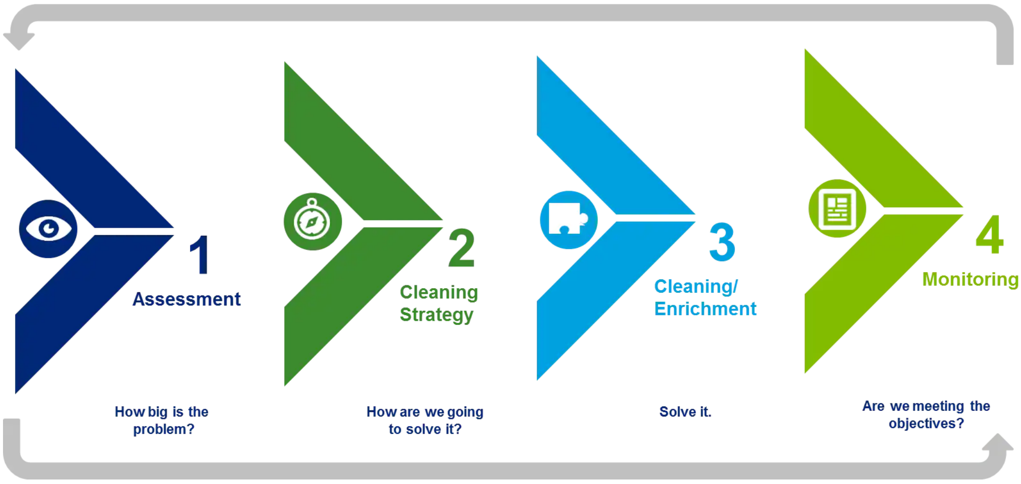


Figure 2.5: Conceptual Data Quality Framework developed by Deloitte Belgium. Source: [33]

This Framework is composed by four steps:

- Assessment: in this step, the Data Quality is evaluated.

- Cleaning strategy: refers to planning to solve the data quality issues.
- Cleaning enrichment: in this step, the methods and techniques for data cleaning are applied.
- Monitoring: the aim of this step is to verify if the cleaned data to meet the expectations of data consumers.

Thus, in general terms, the view of data quality framework is more conceptual and it is used as a helpful map to provide solutions to data with poor quality in knowledge discovery tasks [34, 11]. The purpose of the framework presented in this doctoral thesis is connect more closely with the data analyst and potentially give them suggestions as to which data cleaning algorithms are the most suitable for data quality issues presented in different knowledge discovery tasks.

2.1.3 Ontology

The ontology term provenances from philosophy, where it is concerned with the nature of being and existence [35]. In artificial intelligence (AI) communities of ontologies are widely used and have many definitions; Gruber [36] provided a popular one: an ontology is an “explicit specification of a conceptualization”. The conceptualization represents a specific world view on the domain of interest [37] and it is composed of concepts, attributes, instances and relations between concepts:

- Concept, also called Class, it is a general representation of a group of Individuals that share common features [38].
- Attribute is a feature of a concept [39].
- Instance, also named Individual, it is a specification of a concept [39].
- Relation describes the way in which individuals or concepts relate to each other [38].

Figure 2.6 presents an Ontology example for classification and clustering models. Thus, *Classification* and *Clustering* are subclasses of *Model. Decision tree, Neural Network, Support Vector Machine, Naive Bayes, K Nearest Neighbor* are individuals of *Classification* Class. Besides, *Model* and *Dataset* classes have a relation: “model is built with dataset”.

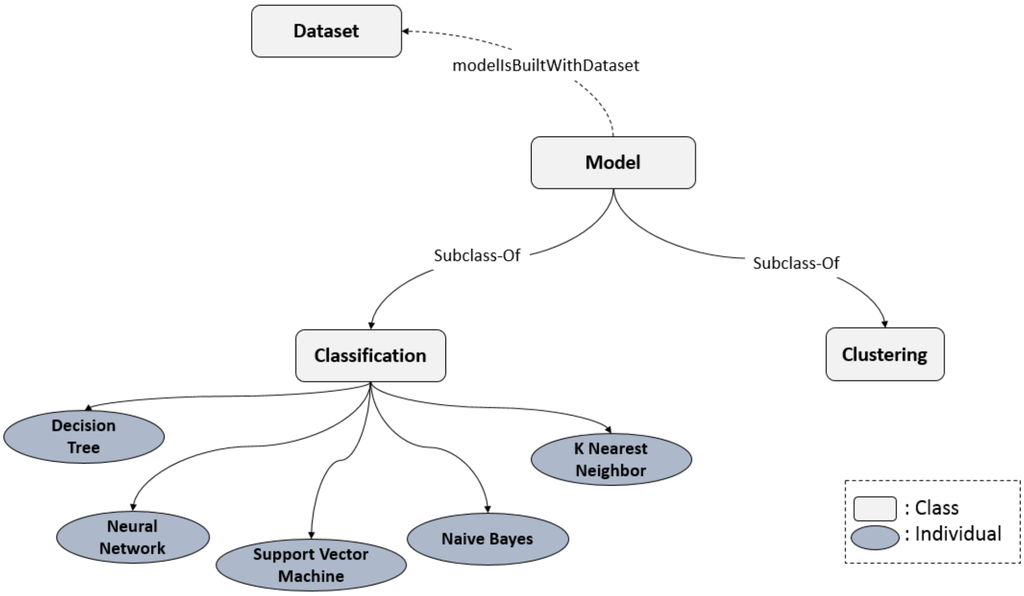


Figure 2.6: Example of Ontology for classification and clustering models. The blue circle represents the class individuals while the gray square depicts the classes. The dotted line represents a relation between two classes and the solid line means a hierarchical relation.

Several languages are used to describe ontologies such as Conceptual Graphs (CG) [40], Description Logics (DL) [41], First Order Logic (FOL) [42] and Ontology Web Language (OWL) [43].

OWL is the most popular language recommended by the W3C [43]. This language represents the ontology elements (classes, attributes, instances, relations, etc.) through different formats as XML, RDF, and RDF Schema (RDF-S) [44]. Thus, OWL has the ability to interpret the available content on the Web [45].

2.1.4 Case-based reasoning systems

Case-based reasoning (CBR) is a type of intelligent system that utilizes the knowledge acquired from past experiences (also they are named cases) to solve a given problem [46, 47, 20]. The main difference of Case-based reasoning from other reasoning techniques is that it does not lead from true assumptions to true conclusions. In other words, if the solution of a past case were correct for its original problem, this may not be the exact solution for a new problem. Therefore, the reuse of the past case may only be "close" to the correct solution of the new problem. This means that applying CBR is a kind of approximate reasoning. In fact, a CBR is essentially centered on retrieval of cases most similar to a new problem [48]. The general architecture of case-based reasoning systems is shown in Figure 2.7.

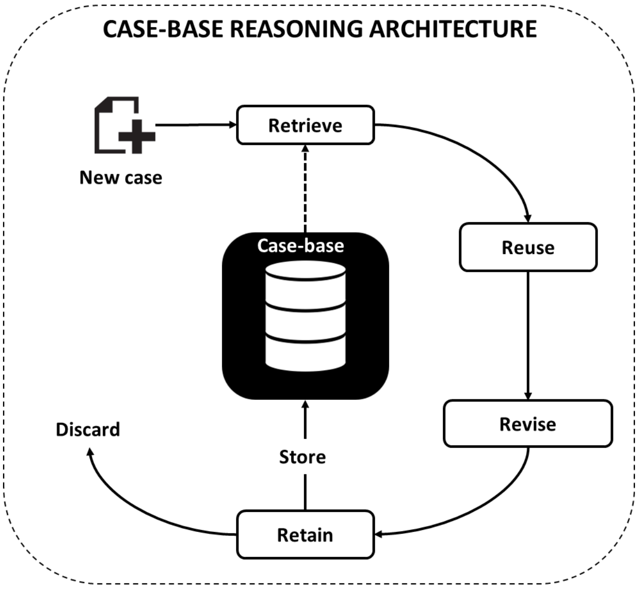


Figure 2.7: The general architecture of case-based reasoning systems.

The cases are composed of a problem and solution and they are stored in a case-base. The CBR cycle is divided in four main steps:

- Retrieve: the purpose of retrieval phase is to search in the case-base a case or a small set of cases similar to the problem or current situation. In other words, the new case C_q is compared with cases of the case-base in order to find the most similar past case C_t. This phase is strongly connected with the case representation and the retrieval techniques. There are many retrieval techniques such as similarity measures and filtering cases [48]. The first consists in computing a similarity score for each case of the case-base and C_q. The second approach selects a set of cases of the case-base respect to similarity criteria of C_q. In our CBR, we proposed a hybrid retrieval mechanism between similarity measures and filtering cases (Subsection 5.2).

- Reuse: the solution of retrieved case C_t is selected as a solution to be reused in C_q. The reuse is simple when the new problem of C_q is equal to the retrieved case problem C_t. In otherwise, the solution of C_t requires an adaptation supported in the knowledge of the domain [48].

- Revise: a solution is proposed (the adapted case in the reuse phase) to solve the new problem of C_q, and it is completed when it is confirmed. Revise aims to evaluate the applicability of the proposed solution in the real world [48]. When the CBR is evaluated in the real world, some aspects may not have considered in the model. This fact is named the frame problem in Artificial Intelligence when all objects of the real world cannot be modeled [49].

- Retain: if the solution of the C_q is successful, then the case new C_q is stored in the case-base to be reused in the future.

For the CBR proposed, we included the Revise step into Reuse phase.

From the general architecture of case-based reasoning systems, the authors of [50] proposed different CBR families:

- Textual CBR: the cases are given as text in natural language [51].
- Knowledge-intensive CBR: a rich domain model is built for supporting small case-bases [50]. In other words, Knowledge-intensive CBR is appropriate when the developers do not have enough experiences available and the knowledge of the domain is represented through of models as ontologies [52].
- Data-intensive CBR: the cases are the main source of information with no domain knowledge available [53].
- Distributed CBR: multi-agent systems collaborate to reach conclusions based on their particular case bases [54].

We built a CBR based on Knowledge-intensive due our case-base is composed of 56 cases. Thus, we proposed a Data cleaning ontology (Chapter 4) for case representation, also in Reuse phase, the Data cleaning ontology suggests similar solutions to the solution space of the retrieval case.

2.2 Related works

This section presents a review of the current literature around three major topic areas. The first section covers related works of frameworks in data quality. Second section concerns related works of ontologies for data cleaning. Finally, third section presents Case-based reasoning systems.

2.2.1 Data Quality Frameworks

Several studies provided data quality frameworks in relational databases, conceptual (theoretical guide process), health systems and big data. Table 2.1 presents a summary of the related works. Most of the works are from the relational databases and data warehouses.

Table 2.1: Related works: Data Quality Frameworks

Works	**Publication year**	**Area**
[55, 28, 56, 57, 58, 59]	2000 – 2015	Databases, Data warehouses
[60, 61, 62]	2002 – 2018	Health systems
[63, 64, 65, 66]	1995 – 2016	Conceptual
[67, 59]	2013 – 2015	Big data

Authors in [55] developed a data quality framework to customer relationship management problem in relational database. The framework is composed by two components:

- Validation design: validates the schema of the input data by: integrity constraints, validation of overloaded table.
- Customer profiling: implements the necessary tables and data quality rules to capture customer preferences as: customer demographic data, and information about a customer's preferences for particular products, areas of interest, and customer activity.

Framework for a quality-driven mining rules is proposed in [28]. The main contributions are: (i) A quality of Data metadata (extension of Common Warehouse Metamodel) which stored data quality measures and cleansing methods description (eliminating duplicates, handling inconsistencies, managing imprecise data, missing data, and data freshness). (ii) A method for scoring the quality of association rules that combines QoD measures. Data quality measures and cleansing methods are computed on SQL.

In [56] offers data cleansing process for relational databases: data transformation, duplicate elimination and data fusion. Each data cleansing process is supported by four type of transformations:

- Mapping: produces records with a suitable format by applying operations such as column splitting.

- Matching: searches pairs of tuples that contain the same real object.
- Clustering: creates groups based on high similarities among real objects and a set of criteria.
- Merging: applies to each individual cluster in order to eliminate duplicates or produce new records for the resulting integrated data source.

The work presented in [66] develops a framework for analyzing data quality research, and uses it as the basis for organizational databases. The framework consists of six elements:

- Management responsibilities: from the requirements of a client/company, data quality policies are defined.
- Research and development: this phase involves the selection of dimensions for assessing the data quality.
- Production: this task analyses the raw data based on set of quality dimensions.
- Distribution: this module organizes the data produced by manufacturing systems.
- Personnel management: this element assesses the data related with personal abilities as training, formal qualification and the motivation.
- Legal function: the aim of this module is to guarantee the data product safety through a traceability system.

DQ^2S is a framework and tool for combining traditional data management with data profiling targeted at data cleansing described in [57]. The framework allows database users to profile their data while querying the database in a declarative way, in preparation for data cleansing, considering dimensions of data quality, such as accuracy, completeness, timeliness and reputation. The quality-related data properties together with the data profiling algorithms represent the criteria under which data is assessed, measured and filtered, in accordance with definitions of data quality dimensions chosen and modeled by the user.

In [63] data quality framework is applied to monitor and improve the content in an e-government meta-data repository, using syntactic, semantic and pragmatic data quality metrics: (i) syntactic refers to validations with respect to a predefined schema and/or set of programmatic rules, (ii) semantic applies to conformance with the immaterial object or real world physical objects the data intends to represent, (iii) pragmatic denotes the users perceived quality of the data.

Framework based on indicators to measure the quality of Open Government Data was defined in [64]. Framework approach to define an open data quality measurement consists of three parts: (i) identification of the most suitable data quality model as theoretical support of the measurement framework e.g. Total data quality management (TDQM), The Data Warehouse Quality methodology (DWQ), Total information quality management (TIQM), (ii) methodology for the selection of data quality characteristics and metrics: completeness, expiration, understandability (iii) results on the selection of data quality characteristics and metrics: incomplete data, out-of-date data, lack of metadata.

Researches in [58] built a framework for data quality management of enterprise data warehouse based on an object-oriented data quality model (OODQM). The data quality requirements (from dimensions: completeness, correctness, usability, currency, consistency, and relevance), the participators, the data quality checking object, and the possible data quality problems, form the core components of OODQM.

Other data quality frameworks are focused in health systems. For instance, [60] proposes a data quality assessment framework to electronic medical record when matching multiple data sources regardless of context or application. The first assessment phase defines variables of interest for matching multiple data sources. The second assessment identifies and assess if the analytical variables of interest are present and sufficiently represented in the multiple data sources to answer the research questions.

The authors in [61] proposed an initial framework for cloud-based health care systems and electronic health record. The process began with gathering data quality dimensions in organizations and health care systems. In this step, literature review and dictionaries were used to avoid dimensions with the same implication. The next step was to check whether the dimension was relevant to electronic health record content and requirements. The resulting dimensions were grouped into three categories considered the main elements of e-health care systems: information, communication and security.

Framework of procedures for data quality assurance in medical registries is proposed in [62]. Procedures in the framework have been divided into procedures for the coordinating center of the registry (central) and procedures for the centers where the data are collected (local). These central and local procedures are further subdivided into (i) causes of insufficient data quality e.g. Illegible handwriting in data source, incompleteness of data source, unsuitable data format in source, (ii) actions to be taken / corrections. A literature review and a case study of data quality formed the basis for the development of the framework.

Other works are used in different domain applications. The authors [65] have identified relationship amongst data quality dimensions while providing primary empirical support to develop a framework for data quality dimensions. Focusing on four significant quality dimensions: accuracy, consistency, timeliness, completeness. A qualitative approach was conducted using a questionnaire (37 surveys) and the responses were assessed to measure reliability and validity of the survey. Factor analysis and Cronbach-alpha test were applied to interpret the results.

In [67] is presented a data quality framework to manage data sources in Enterprise Service Bus (ESB). The framework measures data quality coming from different sensors and selects the most suitable data source among all available data sources, in respect to the data quality metrics: accuracy, trueness, completeness, timeliness, and consistency. The authors validated the data quality framework through wind sensors of a mill. These were located far from the coastline where the weather is harsh, wind sensors are subject to the moisture and corrosion.

The work in [59] proposes a big data pre-processing quality framework, which consists of a data quality management system based on data quality dimensions: accuracy, completeness, and timeliness. It is used for data quality profile generation through data quality rules as: data type, data format, and domain. These rules are applied as pre-processing activities prior to data analysis. The data quality profile selection was evaluated with an electroencephalograph dataset.

2.2.1.1 Shortcomings

We observed a large diversity of data quality frameworks used in the literature designed for relational databases, data warehouses, health systems, and enterprise service bus. However, the related works are not focused in address data quality issues in knowledge discovery tasks. Although [59, 67] are quality frameworks for big data pre-processing these works lack:

- A user oriented process to address orderly many data quality issues (e.g, missing values, outliers, imbalanced classes, mislabeled instances, duplicate instances, high dimensionality).

- Recommendations of the suitable data cleaning algorithm to address data quality issues.

2.2.2 Data Quality Ontologies

From data quality, ontologies have been constructed for several domains as relational databases, health systems, etc. Also ontologies for data mining projects as we can see in Table 2.2.

Table 2.2: Related works: Data cleaning ontologies

Works	Publication year	Area
[68, 69, 70, 71]	2002 – 2012	Databases
[72, 73]	2014 – 2015	Health systems
[74, 75, 76]	2008 – 2016	Others
[77, 78, 79, 80, 81]	2008 – 2018	Data mining projects

Several data quality ontologies proposed in the literature are focused in relational databases. OntoClean, an ontology-based approach to cleaning of databases (DB) is designed in [68]. OntoClean selects data cleaning algorithms respect to the user's goal. The selected data cleaning algorithm is applied to DB based on the results produced from queries on ontology. OntoClean address data quality issues as typographical errors, synonymous record problem, missing data, inconsistent data entry format.

The study carried out in [69] designed a model to represent of data cleaning operations, enabling their reuse in different databases. The model is composed for an orthogonal cleaning ontology and domain ontologies. Operations which are generic and independent of domains are defined in the orthogonal ontology (at the attribute level: missing value in mandatory attribute, syntax and domain violation; at the tuple level: integrity constraint violation) and the dependent ones in the domain ontologies.

Rule mining for automatic ontology based data cleaning is proposed in [70]. This consists of checking tuples for correctness. When invalid tuples are being detected, they have to be modified using valid tuples stored in their ontology. After a learning phase ontology-based user selections are being saved and used to identify replacement rules. The rules are applied automatically when erroneous data is detected.

The work in [71] contains a method for dealing with semantic heterogeneity during the process of data cleaning, which is the difference of terminologies in distinct data sources. They are based on linguistic knowledge provided by a domain ontology in order to generate some correspondence assertions between tuples. These assertions are used during the integration of the data.

Other authors are focused in data cleaning ontologies for health systems. For example, a data quality ontology for electronic health records is developed in [72]. The healthcare data quality literature was mined for the important terms used to describe data quality concepts. These terms were harmonized into a data quality ontology that represents core data quality concepts. Four high-level data quality dimensions was defined: Correctness, Consistency, Completeness and Currency.

The work presented in [73] developed an ontology to assess three data quality dimensions: uniqueness, existence and consistency in patient clinic databases. They are supported in domain ontology to analyze relations as a doctor can not be treated himself as a patient.

Other works use domain ontologies to support data quality issues (e.g, missing values, spelling and format errors, heterogeneity data) such as construction of reservoir models [74], selection of features in datasets related to cancer [75], preparation of genotype-phenotype relationships in a familial hypercholesterolemia dataset [76].

From data mining, authors proposed ontologies for selection of KD algorithms as Knowledge Discovery in Databases Ontology (KDDONTO) [77]. This ontology supports the discovery of KDD web services and the composition of KDD processes. KDDONTO was built based on METHONTOLOGY methodology [82]. The implementation of KDDONTO is formed of 95 classes, 31 relations and more than 140 instances, representing algorithms for classification, clustering, and evaluation.

Ontology Data Mining (OntoDM) [78] has been designed for general purposes. OntoDM includes definitions of basic data mining entities, such as data type and dataset, tasks, algorithms and experiments. This ontology is based on principles of Ontology for Biomedical Investigations (OBI) [83] and generic Ontology of Experiments (EXPO) [84]. The OntoDM ontology defines around 100 classes. All of the classes are extensions of top level classes that correspond and can be easily mapped to OBI and EXPO.

Data Mining OPtimization Ontology (DMOP) [79] has been developed for the automation of algorithm and model selection through semantic meta-mining that makes use of an ontology-based meta-analysis of complete data mining processes in view of extracting patterns associated with mining performance. DMOP contains detailed descriptions of data mining tasks, data, algorithms, and workflows. DMOP was deployed in data mining environment RapidMiner.

Data Mining Ontology (DMO) [80] was designed to support meta-learning for algorithm selection. DMO provides a conceptualization of data mining tasks, methods/algorithms and datasets. Also, DMO considers features of the models as the structure and parameters, the cost function to quantify the appropriateness of a model, and the optimization strategy to find the model parameter values that minimize this cost function. This ontology was developed in OWL2 using the Protegé editor.

The authors in [81] proposed a Big Data integration ontology, where the aim is the data integration process under schema evolution by systematically annotating it with information regarding the schema of the sources. The ontology integrates into a machine-readable format, semi-structured data while preserving data independence regardless of the source formats or schema. This ontology is divided into two levels. The first level (global) provides a unified schema for querying as well as relevant metadata about the attributes, while the second level (source) deals with the physical details of each data source.

2.2.2.1 Shortcomings

The related works presented above conduct data cleaning ontologies. In Table 2.3 are presented the shortcomings of these works.

Table 2.3: Shortcomings of the related works: data cleaning ontologies

Works	**Shortcoming**
[68, 69, 70, 71, 72, 73, 74, 75, 76, 74, 75, 76]	These do not focus on data quality issues in classification or regression tasks.
[77, 78, 79, 80]	They are centered in the selection of KDD algorithms as models of classification, regression and clustering.
[81]	This addresses data integration process in Big data environments.

We present in Section 4 a data cleaning ontology to address data quality issues in classification and regression tasks.

2.2.3 Case-based reasoning systems

The Case-based reasoning systems have received considerable attention by several researchers from different areas as health systems, chemical process, companies, Internet, housing and other fields as shown Table 2.4.

Table 2.4: Related works: case-based reasoning systems

Works	Publication year	Area
[85, 86, 87, 88]	2016 – 2018	Health systems
[89, 90, 91]	2017	Chemical
[92, 93, 94]	2016 – 2018	Companies
[95, 96, 97]	2016 – 2018	Internet
[98, 99]	2017	Housing
[100, 101, 102, 103, 104, 105, 106, 107, 108, 109]	1996 – 2010	Knowledge discovery
[110, 111, 112, 113, 114]	2017 – 2018	Others

The CBR in health systems is used for the diagnosis of different diseases as cancer of breast [88] and gastrointestinal [87], scenarios of depression [85], also the insulin doses for persons with diabetes mellitus [85]. From the chemical area, the CBR is used for fault diagnosis of Tennessee Eastman process [89, 91] and Biochemical oxygen concentration in a Chinese wastewater treatment plant [90]. In the companies, CBR is used of different ways, for example in [93], the CBR estimates the cost of new product development, in [92] the CBR predicts the bankrupt of a company, while in [94], CBR is used for selection of team members. Works applied to Internet are focused on phishing web detection [97] and web service discovery and selection [96], while CBR of [95] is centered in the identification of leaders of specific domains within the on-line communities. From housing area, the research of [98] estimates the construction cost of multi–family housing complexes and the authors of [99] detect risk scenarios in elderly people living alone in a smart homes. CBR's in other fields propose solutions to the problem of traffic congestion [112], diagnosis of railway turnout system [113] and detection of volcano status [114].

From knowledge discovery tasks, authors of [100, 101] proposed two approaches for the recommendation of data mining algorithms through case-based reasoning systems. In [100] a framework is proposed to guide users in KDD tasks. The main goal is reuse task-oriented planning based on Problem Solving Methods (PSM). This method describes how to solve a data mining task by decomposing and defines an order on the subtasks in the decomposition through a controlflow. In [101] built a plug-in for IBM SPSS Modeler named CITRUS. The cases are represented by data mining workflows modeled in IBM SPSS. Based on data mining task description, CITRUS loads the most similar case through hierarchical planner which builds partial workflows from data mining operators.

In [102] the authors proposed an Algorithm Selection Tool (AST) to support the selection of classification and regression models. The case–base contains 80 cases composed by dataset meta–features. Also, AST defines filters based on user preferences, such as whether the produced model is interpretable (true/false) and the relative training and testing time (fast/slow). The algorithm selection is a decision based on application restrictions (top-down), a given dataset with its meta-data characteristics (bottom-up) and on knowledge about the available algorithms.

The MiningMart project [103] aims at reuse of successful preprocessing practices (discretization, handling of null values, aggregation of attributes into a new one, collecting of sequences from time-stamped data) in SQL databases. A meta-data model named M4 is used to define all steps of preprocessing chain and all the data involved. MiningMart describes all cases in an ontology with informal annotations, such as the goals and constraints of each problem.

The authors of [104, 105, 106, 107] built a CBR based on the CRISP-DM phases. The first work [104] exposed the design considerations of the CBR through the concept: data mining Assistant. The second work [106] presented a proposal of hybrid Data Mining Assistant, based on the CBR paradigm and the use of an ontology, in order to provide additional assistance (i.e. by means of recommendations and heuristics) to a user during the various phases of the Data Mining process. The ontology of the CBR is built in the third work [106]. This ontology was implemented in Web Ontology Language-Description Logic (OWL-DL) using the Protégé software tool [115]. The ontology contains approximately 200 data mining concepts of the CRISP-DM methodology. Finally, in the last work [107], the authors built the CBR based on expert rules expressed in SWRL, which are stored in the ontology mentioned above. The cases are represented by dataset meta-features as number of examples, attributes and classes, mean kurtosis, mean skewness, etc. K-nearest neighbor and arithmetic similarity function were used as retrieval mechanism. The CBR system returns two scores: one based on similarity and the other based on user satisfaction. After a case has been selected, the proposed system guides the user through practices of five phases of CRISP-DM methodology (business understanding, data preparation, modeling, and evaluation).

In [110] built a CBR for data preparation in electronic diabetes records. The paper is concentrated on data preprocessing of missing values, feature selection, feature weighing, outlier detection, and normalization. These steps are performed sequentially on the raw case-base data to produce a new high quality case base. They have 60 case-base and K-nearest neighbor algorithm with local-global approach is used for case retrieval. GapIt is a user-driven case-based reasoning tool for infilling gaps in daily mean river flow records [111]. It was tested in the gauging network of Luxembourg to perform gap infilling on daily values. Given a set of flow time series, GapIt builds a database of artificial gaps for which it computes several flow estimates, to find the best combinations of infilling algorithm and automatically selected donor station(s), according to state-of-the-art performance indicators.

A similar approach presented in [108] uses data mining ontologies combined with the CRISP-DM methodology to advise the suitable application of CRISP-DM tasks in data mining projects. It also uses the rules stored in ontologies. Unfortunately, there are many missing details about this approach.

The authors of [109] developed a data mining assistant for selection of classification model. The retrieval mechanism is based on k-nearest neighbor. Unfortunately, this work lack of details about its approach.

2.2.3.1 Shortcomings

Previous works of CBR are focused in different fields. The works of knowledge discovery tasks are directly related to our research (recommendation of data mining algorithms). Table 2.5 presents the shortcomings of the knowledge discovery works.

Table 2.5: Shortcomings of the related CBR works

Works	Shortcoming
[100, 101]	The works return partial or abstract workflows, leaving it to the user to incomplete guided process.
[102],[108],[109]	These works recommend the suitable classifier.
[103]	This work is focused in data quality issues of SQL databases
[104, 105, 106, 107]	The works suggested general recommendations in the phases: business understanding, data preparation, modeling, and evaluation of the CRISP-DM methodology.
[110, 111]	The works proposed data cleaning solutions for specific domain (records: diabetes and river flow).

We observed a large diversity of CBR systems in the literature, however the CBR for knowledge discovery tasks are not focused on recommending the suitable data cleaning algorithms for classification or regression tasks. In Chapter 5 we propose a case-base reasoning to recommend the suitable data cleaning methods in classification and regression tasks. The CBR proposed supports each task of the conceptual framework for guide to user in the analysis of data quality issues proposed in Chapter 3.

2.3 Summary

In this chapter, we explained the most relevant concepts to understand the thesis contributions. First, we described the methodologies for knowledge discovery (KD) from data as *Knowledge Discovery in Databases (KDD)* [116], *Cross Industry Standard Process for Data Mining (CRISP-DM)* [13], *Sample, Explore, Modify, Model and Assess (SEMMA)* [23] and *The Data Science Process* [117]. Subsequently, we presented the concepts: *Data quality framework*, *Ontology*, and *Case–based Reasoning*. For each one of these concepts, we made a review of the current literature and we found the next shortcomings:

- **Data quality frameworks:** the related works are not focused in address data quality issues in classification or regression tasks. Although [59, 67] are quality frameworks for big data pre-processing these works lack:
  - A user oriented process to address orderly many data quality issues (e.g, missing values, outliers, imbalanced classes, mislabeled instances,

    duplicate instances, high dimensionality).

  - Recommendations of the suitable data cleaning algorithm to address data quality issues.

- **Data Quality Ontologies:** the related works do not focus on data quality issues in classification or regression tasks.

- **Case-based Reasoning systems:** the CBR for knowledge discovery tasks are not focused on recommending the suitable data cleaning algorithms for classification or regression tasks.

3. Conceptual Data Quality Framework

This chapter presents the conceptual framework to address poor quality data in classification and regression tasks. The methodology "*Building a Conceptual Framework: Philosophy, Definitions, and Procedure*" [21] was adapted to build the proposed process. This offers an organized procedure of theorization for building conceptual process. The advantages of use this methodology are the flexibility for make modifications, and the easy understanding. Below are explained the adapted phases for building the conceptual framework for data cleaning in classification and regression tasks.

3.1 Mapping the selected data sources

The first phase identifies the data quality issues to classification and regression tasks. This process includes review texts and other sources of data as research papers, standards or methodologies. From knowledge discovery we found four relevant methodologies (Explained in subsection 2.1.1): *Knowledge Discovery in Databases (KDD)* [116], *Cross Industry Standard Process for Data Mining (CRISP-DM)* [13], *Sample, Explore, Modify, Model and Assess (SEMMA)* [23] and *The Data Science Process* [117]. Table 3.1 shows the data quality issues considered in the KDD methodologies.

Noise, *missing values*, *outliers*, and *high dimensionality* were the data quality issues found in the knowledge discovery methodologies presented in Table 3.1 [118, 119, 13, 23, 117, 116].

Table 3.1: Data quality issues considered in data mining and machine learning methodologies

Methodology	**Methodology Phase**	**Data Quality Issue**
KDD	Preprocessing	Noise
		Missing Values
		Outliers
		High Dimensionality
CRISP-DM	Data Understanding	Missing Values
	Data Preparation	Outliers
		High Dimensionality
SEMMA	Explore and Modify	Outliers
		High Dimensionality
Data Science Process	Clean Data and Exploratory data analysis	Missing Values
		Duplicates
		Outliers

In addition, the authors of [120] proposed a taxonomy of data quality challenges in empirical software engineering (ESE), based on an literature review. The ESE taxonomy captures data quality issues presented in empirical software engineering, although some of the data quality issues of the taxonomy are not peculiar to ESE data sets. Besides the data quality issues previously found (Table 3.1), in the work of [120] we found new data quality issues as *Inconsistency*, *Redundancy*, *Amount of data*, *Heterogeneity*, and *Timeliness*.

Finally, we reviewed papers where the data quality issues (previously mentioned) are addressed. We reviewed research papers from IEEE Xplore, Science Direct, Springer Link, and Google Scholar [118]. Table 3.2 shows the papers found by data quality issue and informational source.

Table 3.2: Number of papers found to address data quality issues [118].

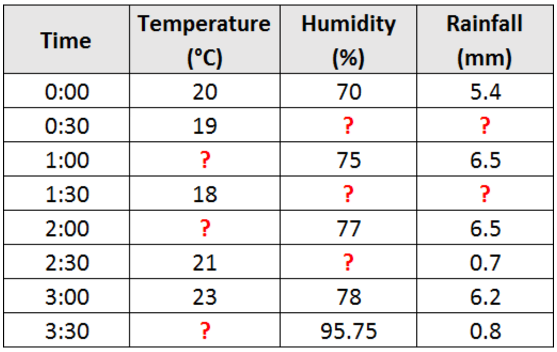
	Number of papers				
Data quality issues	**IEEE Xplore**	**Science Direct**	**Springer Link**	**Google Scholar**	**Total**
Redundancy	24	13	10	8	55
Amount of data	23	15	10	5	53
Outliers	28	10	7	2	47
Missing values	21	14	4	0	39
Heterogeneity	11	3	1	18	33
Noise	15	2	2	0	19
Inconsistency	9	5	0	2	16
Timeliness	2	0	1	1	4

According to papers found in the Table 3.2, the redundancy is refereed to: high dimensionality and duplicate instances, and the amount of data to imbalanced class. Data quality issues as missing values, outliers, amount of data, and redundancy have received greater attention from research community (papers found: 39, 47, 53 and 55 respectively). Meanwhile noise (17 papers) have less attention because it is defined as general consequence of the data measurement errors.

3.2 Understanding the selected data

The aim in this phase is understand the data quality issues from classification and regression tasks. Next, we present a description of each data quality issue.

- **Noise**: defined by [121] as irrelevant or meaningless data. The data noisy reduce the predictive ability in a classification and regression models [122].
- **Missing values**: refers when one variable or attribute does not contain any value. The missing values occur when the source of data has a problem, e.g, sensor faults, faulty measurements, data transfer problems or incomplete surveys [123].
  Considering the data collected by weather stations, some values are missed due to lapses found in the sensors, electrical interruptions, and losses in the data transmission, etc. In Figure 3.1 we present a dataset of weather stations with missing values represented by symbol “?”.



Time	Temperature (°C)	Humidity (%)	Rainfall (mm)
0:00	20	70	5.4
0:30	19	?	?
1:00	?	75	6.5
1:30	18	?	?
2:00	?	77	6.5
2:30	21	?	0.7
3:00	23	78	6.2
3:30	?	95.75	0.8

Figure 3.1: Example of missing values generated by weather stations for Temperature, Humidity and Rainfall. The columns represent the dataset attributes. The rows represent the dataset instances with sampling frequency of 30 minutes. The symbol ? in red color represents the missing values in the dataset.

- **Outliers**: can be an observation univariate or multivariate. An observation is denominated outlier when it is deviated markedly from other observations, in other words, when the observation appears to be inconsistent respect to the remainder of observations [124, 125, 126].
  In Figure 3.2, we show an example of outliers (red points) presented in a dataset for house cost prediction (price in 1000 of US dollars) based on area built of the house (square meters).

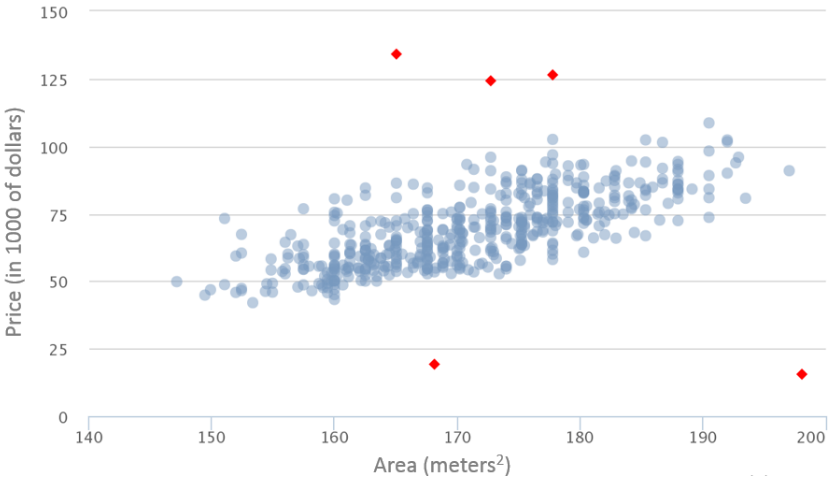


Figure 3.2: Outliers in house cost prediction. The dots in red color represent the outliers respect to remaining of data represented by blue dots.

- **High dimensionality**: is referred when dataset contains a large number of

features [127]. With the presence of a large number of features, a learning model tends to overfit, resulting in their performance degenerates [128, 129].
For example, in genetic field, the number of features can exceed the number of instances [130]. The Microarrays (measure gene expression), contain thousands of observations, and each observation contains large number of genes [131].

- **Imbalanced class**: is considered when a data set exhibits an unequal distribution between its classes [132]. When a dataset is imbalanced, the approximation of the misclassification rate used in learning system can contribute negatively to decrease the accuracy and the quality of learning [133].
  Figure 3.3 shows a dataset with an imbalanced class: loan approval, where the red stars represent the instances (14) with the positive decision of loan, while the gray circles represent the instances (33) with the negative decision of loan.

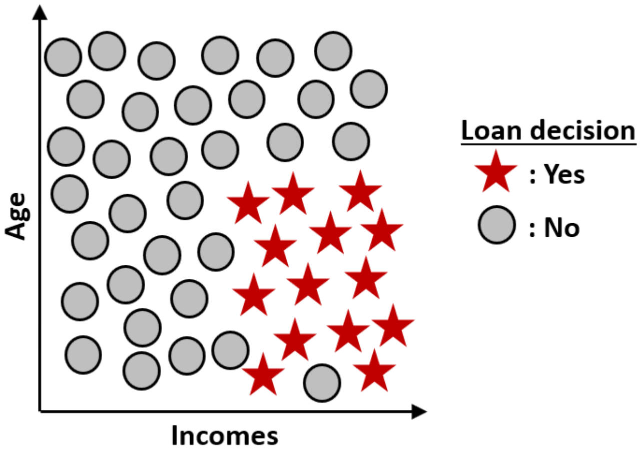
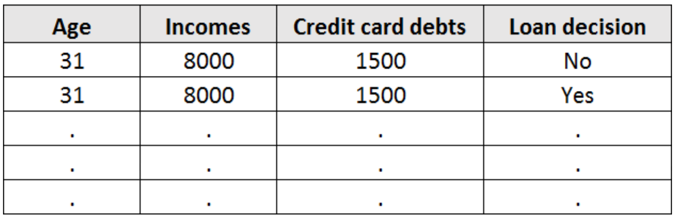


Figure 3.3: Imbalanced class in a dataset for loan approval. Red stars represent the instances with the positive decision of loan. Gray circles represent the instances with the negative decision of loan.

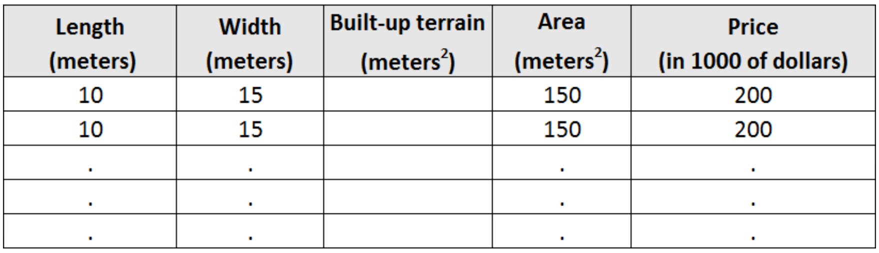
- **Inconsistency**: refers to a lack of harmony between different parts or elements of the dataset; instances that are self-contradictory (duplicate samples have different class labels), or lacking in agreement when it is expected [120].
  Assuming we have a dataset for loan approval composed of three attributes: Age, Incomes, and Credit card debts of a person, and the class: Loan decision as show Figure 3.4. The instances 1 and 2 present inconsistency due to duplicate values of the attributes and the Loan decision is different.



Age	Incomes	Credit card debts	Loan decision
31	8000	1500	No
31	8000	1500	Yes
.	.	.	.
.	.	.	.
.	.	.	.

Figure 3.4: Inconsistency of a dataset for loan approval. The columns represent the dataset attributes. The rows represent the dataset instances. The inconsistency is presented in the instances 1 and 2 due to attributes have the same values but the values of the class are different.

- **Redundancy**: represents duplicate instances and redundant attributes in datasets which might detrimentally affect the performance of models [120]. For example, Figure 3.5 depicts a dataset for house cost prediction (price in 1000 of US dollars) with attributes as Length (meters), Width (meters) and Area (square meters) of a house. A case of redundant is presented by the attributes Length and Width because these attributes represent the same information of the Area and Built-up attributes. In addition, the attributes Area and Built-up terrain are redundant due these attributes are the same with different names. In case of duplicate records are illustrated through instances 1 and 2 of the Figure 3.5.



Length (meters)	Width (meters)	Built-up terrain (meters^2)	Area (meters^2)	Price (in 1000 of dollars)
10	15		150	200
10	15		150	200
.	.		.	.
.	.		.	.
.	.		.	.

Figure 3.5: Redundancy of a dataset for house cost prediction. The columns represent the dataset attributes. The rows represent the dataset instances. The redundancy is presented in the attributes Area and Built-up terrain. The duplicate records are depicted in instances 1 and 2.

- **Amount of data**: the amount of data available for model building contributes to the possible statistical significance of generated models converting it in another factor of relevance in the data set construction [120]; small data sets build inaccurate models.
  A real case is presented in [134, 135]. The dataset includes 147 instances to estimate the incidence of rust (values between 0%–100%) in coffee crops.

Nevertheless, the main drawback of these works is the low number of instances to try to predict a continuous value (incidence of rust); if the available examples are few, the dataset does not represent a sample trustworthy of the population, then the classifiers will be not inaccurate [136].

- **Heterogeneity**:defined as incompatibility of information. We distinguish two types of heterogeneity: syntactic heterogeneity refers to differences among definitions, such as attribute types, formats, or precision, while semantic heterogeneity refers to differences or similarities in the meaning of data [137].
  Practical examples are the data collected by weather stations (WS) as show Figure 3.6. Let us suppose that exist two WS with data temperature. The WS "A" measures the temperature with a dot as decimal separator and the WS "B" with a comma. When we try to fuse the temperature data of WS "A" and "B" we find a syntactic heterogeneity issue. Equally, the WS "A" measures the temperature in Celsius degree and the WS "B" in Fahrenheit scale, in this case, we find a semantic heterogeneity issue.

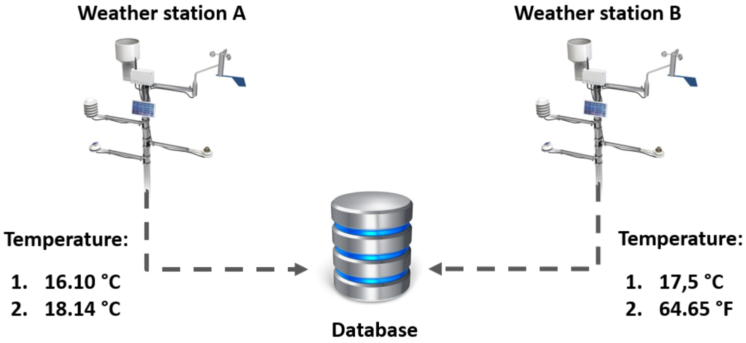


Figure 3.6: Data recollection of temperature by two WS. The WS "A" measures the temperature in different format/scale than WS "B".

- **Timeliness**: has been defined as the degree to which data represent reality from the required point in time [7]. When the state of the world changes faster than our ability to discover these state changes and up-date the data repositories accordingly, the confidence on the validity of data decays with time [138]. e.g., people move, get married, and even die without filling out all necessary forms to record these events in each system where their data is stored [139].
  An example of timeliness is the construction of a classifier for estimation of the rust incidence in coffee crops based on weather data from 1998. The classifier will be accurate to estimate coffee rust in the year 1998; however,

today the classifier does not work due to weather changes occurred in the last years [118].

3.3 Identifying and categorizing components

The aim in this phase is organize and filter the data quality issues according to their meaning. The following changes have been made:

- *Inconsistency*, *Redundancy* and *Timeliness* were renamed as *Mislabelled class*, *Duplicate instances* and *Data obsolescence* respectively to represent better the data quality issues in classification and regression tasks.
- According to the *noise* definition “irrelevant or meaningless data”, we considered kinds of *noise*: *missing values*, *outliers*, *high dimensionality*, *imbalanced class*, *mislabelled class* and *duplicate instances*
- *amount of data*, *heterogeneity* and *data obsolescence* are issues of recollection data process. These data quality issues were classified in a new category called *Provenance*, defines by Oxford English Dictionary as “the fact of coming from some particular source or quarter; origin, derivation”.

Figure 3.7 presents the categories of the data quality issues for classification and regression task. The conceptual framework is focused on solve noise problems in the data.

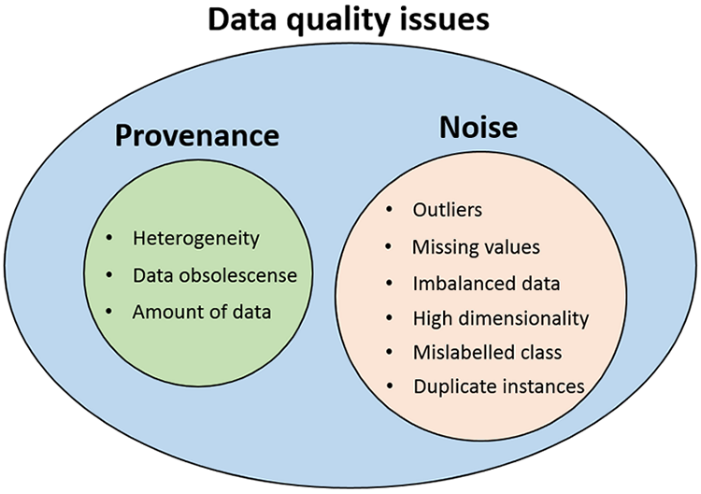


Figure 3.7: Categories of the data quality issues for classification and regression task

3.4 Integrating components

In this phase, first, we define the data cleaning tasks. Subsequently, we propose a cleaning task as a solution for each noise issue. Table 3.3 shows the data cleaning tasks.

Table 3.3: Data cleaning tasks

Noise Issue	**Data cleaning task**
Missing values	Imputation
Outliers	Outlier detection
High dimensionality	Dimensionality reduction
Imbalanced classes	Classes balancing
Mislabelled class	Label correction
Duplicate instances	Remove duplicate instances

- **Imputation**: replaces missing data with substituted values. In the literature were found, four relevant approaches to imputing missing values:
  - *Deletion:* excludes instances if any value is missing [140].
  - *Hot deck:* missing items are replaced by using values from the same dataset [141].
  - *Imputation based on missing attribute:* assigns a representative value to a missing one based on measures of central tendency (e.g, mean, median, mode, trimmed mean) [142].
  - *Imputation based on non-missing attributes:* missing attributes are treated as dependent variables, and a regression or classification model is performed to impute missing values [143].
- **Outlier detection**: identifies candidate outliers through approaches based on *Clustering* (e.g, Density-based spatial clustering of applications with noise - DBSCAN) or *Distance* (e.g, Local Outlier Factor - LOF) [144, 145, 146].
- **Dimensionality reduction**: reduces the number of attributes finding useful features to represent the dataset [147]. A subset of features is selected for the learning process of the regression model [127]. The best subset of relevant features is the one with least number of dimensions that most contribute to learning accuracy [129]. The reduction of dimensionality can be done from four approaches:

- *Filter:* selects features based on discriminating criteria that are relatively independent of the regression (e.g. correlation coefficients) [129].
- *Wrapper:* based on the performance of regression models (e.g. error measures) are maintained or discarded features in each iteration [148].
- *Embedded:* the features are selected when the regression model is trained. The embedded methods try to reduce the computation time of the wrapper methods. [149].
- *Projection:* looks for a projection of the original space to space with orthogonal dimensions (e.g. principal component analysis) [150].

- **Classes balancing**: distributes instances equitable per class. Classes balancing consists of two approaches:
  - *Oversampling*: creates new observations from minority class (e.g. SMOTE: synthetic minority over–sampling technique) [151, 152].
  - *Undersampling*: eliminates instances from majority class (e.g. Tomek link) [153, 152]
- **Label correction**: this data cleaning task identifies instances with the same values in the attributes. If classes are different, the label is corrected or the instance is removed [154].
- **Remove duplicate instances**: identifies and removes duplicate instances [155].

The integration of the data cleaning tasks is depicted in Figure 3.8:

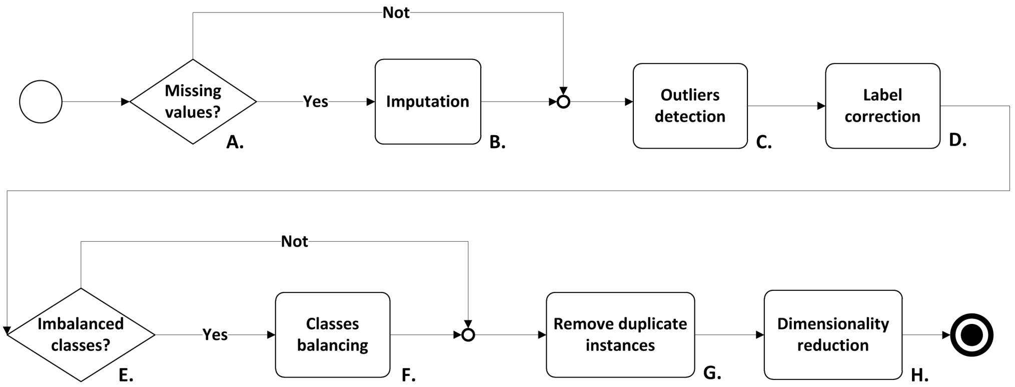


Figure 3.8: Conceptual framework

Thus, a user of the conceptual framework follow the next steps:

A *Verify if dataset contains missing values*: usually missing data are represented by special characters as ?, *, blank spaces, specials words as NaN, null, etc. The first step is to know how the data cleaning algorithm represent the missing values and convert these missing values to same format.

B *Apply imputation algorithms*: once prepared the format of missing values, an imputation algorithm is used. The added values must be verified because the imputation algorithm can creates outliers.

C *Apply outliers detection algorithm*: the outlier detection algorithm finds candidate outliers in the raw dataset or generated by imputation methods.

D *Label correction*: searchers mislabelled instances in the raw dataset or generated by imputation methods. This task is applied for classification datasets.

E *Verify if dataset is imbalanced*: commonly the Imbalance Ratio (IR) is used to measure the distribution of the classes:

$$IR = \frac{Class^{+}}{Class^{-}}$$

Where $Class^{+}$ represents the size of the majority class and $Class^{-}$ the size of the minority class. A dataset with *IR* 1 is perfectly balanced, while datasets with a higher *IR* are more imbalanced [156].
In case of the class has more than 2 labels, Normalized Entropy is used [157]. This measure indicates the degree of uniformity of the distribution of class labels. Denoted by

$$H(Class) = -\sum_{i=1}^{n} q_i log_2(q_i)$$

Where $q_i = p(class = x_i)$ is the probability that $class$ assumes the ith value x_i, for $i = 1, ..., n$. We suppose that each label of the class has the same probability of appearing, therefore the theoretical maximum value for the entropy of the class is $log_2(n)$. Thus the normalized entropy can be computed as:

$$H(Class) = -\sum_{i=1}^{n} \frac{q_i log_2(q_i)}{log_2(n)}$$

The class is balanced when $H(Class)$ is close to 1.
The verification of imbalanced class is used only for classification datasets.

F *Apply algorithm to balanced classes*: this kind of algorithms generates synthetic instances (oversampling techniques) to balance the classes. This task is applied for classification datasets.

G *Apply algorithms to remove duplicate instances*: searches duplicate instances in the raw dataset or generated by algorithms for balance of classes.

H *Apply algorithm for dimensionality reduction*: this kind of algorithms reduce the number of attributes. The attributes retained keep high correlation among themselves.

The conceptual framework proposed is oriented as to how the user address data quality issues in classification and regression tasks. In Chapter 4 is built an ontology to represent data cleaning tasks.

3.5 Validating the conceptual framework

The conceptual framework was tested through 48 datasets (28 datasets for classification and 20 for regression) of the UCI Repository of Machine Learning Databases [158] of the last twenty years (1998 – 2018). The process for testing the conceptual framework (CF) consists of three steps:

1. The UCI datasets are cleaned by our conceptual framework (CF).
2. The cleaned datasets by our conceptual framework (CF) are used to train the same algorithms proposed by authors of UCI datasets.
3. We compare the performance measures (i.e. for classification: *Precision*, *Area Under Curve* and regression: *Mean Absolute Error*) of the models trained with the datasets produced by the authors versus the models trained with the datasets processed by our conceptual framework.

With aim to demonstrate the use of CF, in the subsections 3.5.1.1 and 3.5.2.1, we present the application of the CF in two UCI datasets (from 48 selected datasets). We selected these datasets due are the most largest. For classification tasks, we present the dataset Physical activity monitoring [159], while for regression tasks the dataset related with comments prediction in Facebook posts [160]. Subsequently, we show the performance measures of the models trained with: (i) authors of UCI datasets and (ii) the UCI datasets cleaned by CF.

3.5.1 Classification tasks

3.5.1.1 Test of conceptual framework: physical activity monitoring

The domain physical activity monitoring contains 9 datasets [159]. Each dataset represents one subject (8 males and 1 female). The entire dataset contains 54 attributes and 2.871.916 instances related with sensors measurements (located at chest, hand and ankle). The class has 12 labels: walking, running, cycling and nordic walking, lying, sitting, standing, ascending and descending stairs, ironing, vacuum cleaning and rope jumping. Table 3.4 shows the instances by subject:

Table 3.4: Number of instances of activity monitoring dataset. Each dataset is represented by a subject (Subject 101, ... , Subject 109)

Subjects	**Instances**
Subject 101	376.383
Subject 102	446.908
Subject 103	252.805
Subject 104	329.506
Subject 105	374.679
Subject 106	361.746
Subject 107	313.545
Subject 108	407.867
Subject 109	8.477

- *Imputation*: first we observed how the missing values are distributed on the dataset. Figure 3.9 illustrates the frequencies of missing data patterns. Magenta color shows the missing values and blue color non-missing data. Each row represents a missing data pattern. For example, the first row (bottom up) indicates that *heart rate* has 0.9% missing values when the remaining attributes has data, the sixth row the attributes *temp hand*, *X3D accel hand*, *scale hand*, *resolution hand*, *X3D accel hand 2*, *scale hand 2*, *resolution hand 2*, *X3D giro hand 1*, *X3D giro hand 2*, *X3D giro hand 3*, *X3D magno hand 1*, *X3D magno hand 2*, *X3D magno hand 3*, *orienta hand 1*, *orienta hand 2*, *orienta hand 3*, *orienta hand 4* has 0.004% missing values while the remaining attributes has data.

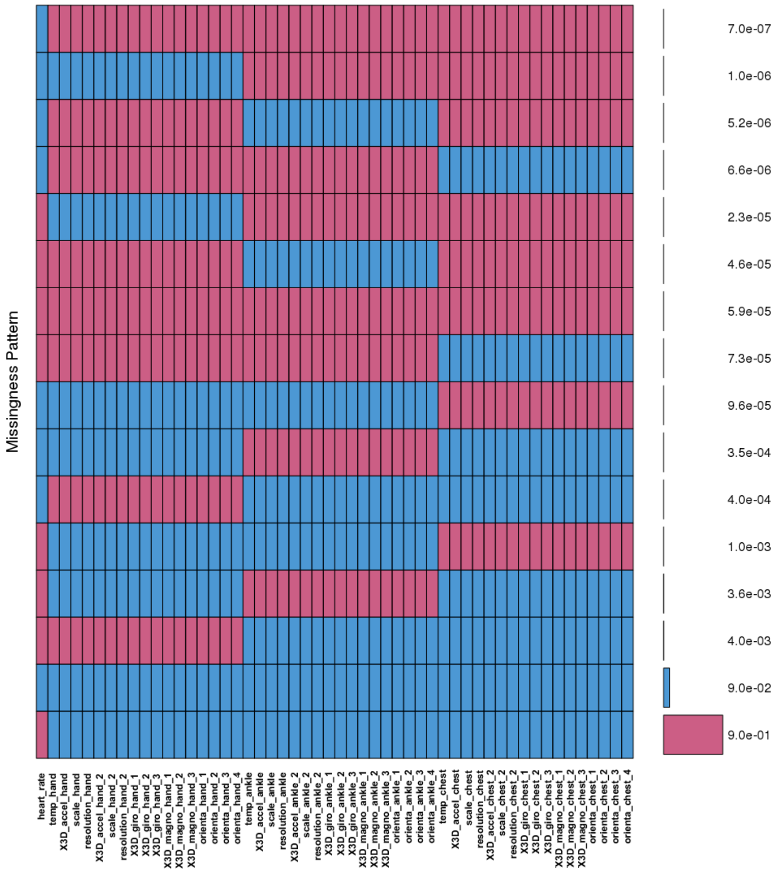


Figure 3.9: Frequencies of missing data patterns. Magenta color shows the missing values and blue color non-missing data

The datasets have around 1.83% – 2.10% of missing values. *Heart rate* is the attribute with highest missing data (greater than 90 %). Thus, we used *List Wise Deletion* to remove *heart rate* attribute and 34 instances. Subsequently, we imputed each subject dataset with *Linear and Bayesian regression*.

- *Outliers Detection*: once imputed values, outliers detection task is applied with the aim to find erroneous imputations. We used *Local Outlier Factor (LOF)*. Table 3.5 shows the potential outliers for each subject. Thus the instances with a *Local Outlier Factor* less than the lower limit or greater than the upper limit are considered potential outliers.

Table 3.5: Potential outliers. The lower and upper limits are calculated by Tukey Fences. Each dataset is represented by a subject (Subject 101, ... , Subject 109)

Subjects	**Potential outliers**	**Lower limit**	**Upper limit**
Subject 101	50.961	0.956	1.059
Subject 102	38.454	0.878	1.203
Subject 103	20.706	0.884	1.191
Subject 104	27.618	0.881	1.198
Subject 105	32.607	0.888	1.182
Subject 106	31.079	0.873	1.214
Subject 107	25.329	0.879	1.204
Subject 108	34.068	0.876	1.209
Subject 109	830	0.875	1.206

The lower and upper limits are calculated by *Tukey Fences* [161]; potential outliers are values below $Q_1 - 1.5(Q_3 - Q_1)$ (lower limit) or above $Q_3 + 1.5(Q_3 - Q_1)$ (upper limit). Where Q_1 and Q_3 are first and third quartiles. In Figure 3.10 the whiskers of the box plots represent the *Tukey Fences* of the *Local Outliers Factor*.

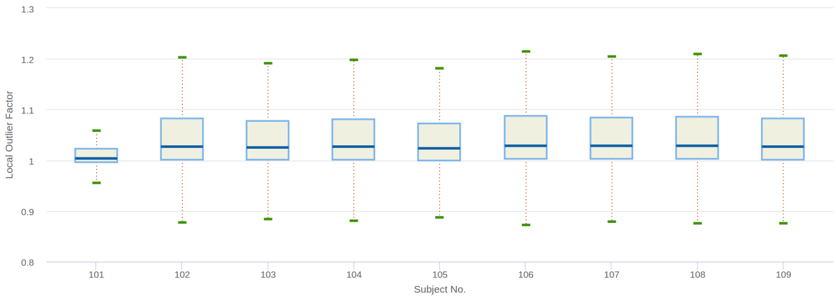


Figure 3.10: Box plot of Local Outliers Factors. Each box plot corresponds to dataset of physical activity monitoring

We removed the potential outliers detected by *Local Outlier Factor* (Table 3.5) which can be erroneous observations generated in imputation task.

- *Label correction:* to correct the labels of the classes, we used *Contradictory instances detection*. The dataset has not contradictory instances.

- *Classes balancing*: we used the balanced classes task for each subject. We used *Synthetic Minority Over-sampling Technique (Smote)*. The dataset has 12 classes, first we identify the majority class and the minority classes, thus we applied Smote for each minority class when $2 < IR < 10$. Figure 3.11 shows the instance distribution per class for all subjects. Purple bars represent the imbalanced dataset, and blue bars the balanced dataset using Smote.

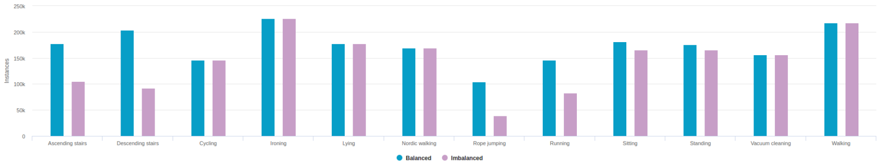


Figure 3.11: Instance distribution per class: balanced vs imbalanced

Smote algorithm increases instances of the classes: *ascending_stairs* (72.199), *descending_stairs* (111.366), *rope_jumping* (64.925),*running* (62.899), *sitting* (16.248) and *standing* (10.683). The remaining classes maintain the same number of instances.

- *Remove duplicate instances*: to detect duplicate instances we used *Standard Duplicate Elimination*. The dataset has no duplicate instances.

- *Dimensionality reduction*: we joined the 9 subjects in one dataset, then we applied dimensionality reduction task. We used *Pearson Correlation* method, the algorithm found weights of continuous attributes based on their correlation with the class. Figure 3.12 presents Top-15 of attributes with highest correlation.

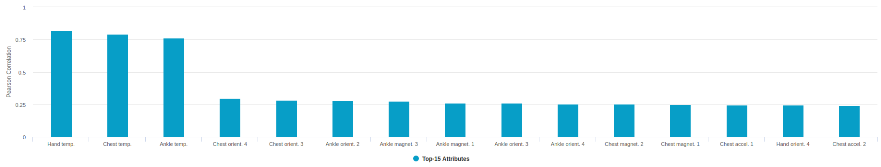


Figure 3.12: Top-15 of attributes with highest correlation for *Pearson* method.

temp hand, *temp chest* and *temp ankle* are the attributes with correlation coefficient greater than 0.75. The correlation values of the remaining Top-15 attributes are among 0.29 - 0.24. The remaining attributes out of the top-15 measure are accelerometers, orientations and magnetometers with correlations among 0.23 - 0.22. We use all attributes taken into account our inexperience in the activity monitoring domain, besides correlation coefficients are different to zero.

- *Results*: authors of Physical Activity Monitoring (PAM) dataset [159] used the classifiers: *Decision tree (C4.5)*, *Boosting - C4.5 decision tree*, *Bagging - C4.5 decision tree*, *Naive Bayes* and *K nearest neighbor* from Weka toolkit. We used the same experimental configuration proposed by the authors [159] based on standard x-fold cross-validation. We do not use a statistical significance test due to the datasets (original and cleaned by CF) are different. The datasets differ mainly in the number of instances and attributes because we used several data cleaning tasks. Table 3.6 shows the accuracy for Physical Activity Monitoring (PAM) dataset.

Table 3.6: Standard 9-fold cross-validation - Accuracy

Classifier	**Physical Activity Monitoring**	**Conceptual Framework**
Decision tree (C4.5)	95.54	99.30
Boosted C4.5 decision tree	99.74	99.99
Bagging C4.5 decision tree	96.60	99.60
Naive Bayes	94.19	77.00
K nearest neighbor	99.46	99.99

In standard 9-fold cross-validation (Table 3.6), our conceptual framework obtained better accuracy in the models: *Decision tree* (99.3%), *Boosted* (99.99%), *Bagging* (99.6%) and *K nearest neighbor* (99.99%), while Physical Activity Monitoring in *Naive Bayes* (94.19%). A systemic problem with *Naive Bayes* is that features are assumed to be independent [162]. An Initial assumption of the results obtained by our approach using *Naive Bayes* is that many attributes represent similar information (e.g, 2 accelerometers for a wrist with 3-axis in two scales = 12 attributes).

3.5.1.2 Comparative study

As mentioned in subsection 3.5, the CF was tested with 28 datasets coming from UCI Repository of Machine Learning Databases [158] for classification tasks. We used the same classifiers proposed by the dataset authors: *Linear Discriminant Analysis (LDA)*, *Random Forest (RF)*, *C4.5 Decision Tree*, *Bagging* and *Boosting* with *C4.5* as base classifier, *Classification and Regression Trees (CART)*, *Support Vector Machine (SVM)* and *Multi Layer Perceptron (MLP)*. Table 3.7 presents two classifiers for each UCI dataset.

Table 3.7: Precision and AUC of the classifiers processed by conceptual framework (CF) and datasets authors of UCI repository. The underlined values represent the highest Precision and AUC (between the classifiers processed by CF and datasets authors).

Dataset	Ref	Approach	Model	Measure	Value %
1.Anuran families calls	[163, 164, 165]	CF	MLP	Precision	97.60
		Authors	MLP		99.00
2.Anuran species calls	[163, 164, 165]	CF	MLP	Precision	98.90
		Authors	MLP		99.00
3.Autism in adolescent	[166]	CF	RF	Precision	99.80
		Authors	RF		91.40
4.Autism in adult	[166]	CF	C4.5	Precision	99.10
		Authors	C4.5		89.80
5.Autism in child	[166]	CF	RF	Precision	99.70
		Authors	RF		85.60
6.Breast tissue detection	[167]	CF	LDA	AUC	92.20
		Authors	LDA		87.30
7.Cardiotocography	[168]	CF	C4.5	Precision	98.60
		Authors	C4.5		97.60
8.Default of credit card	[169]	CF	KNN	AUC	83.60
		Authors	KNN		68.00
9.Human activity recog.	[170]	CF	SVM	Precision	98.40
		Authors	SVM		92.40
10.Ozone level 1 hour	[171]	CF	Bagging	Precision	94.10
		Authors	Bagging		18.50
11.Ozone level 8 hours	[171]	CF	Bagging	Precision	91.30
		Authors	Bagging		41.60
12.Phishing detection	[172]	CF	CART	Precision	83.80
		Authors	CART		90.00
13.Office occupancy	[173]	CF	RF	Precision	99.25
		Authors	RF		98.06

Table 3.7: Precision and AUC of the classifiers processed by conceptual framework (CF) and datasets authors of UCI repository. The underlined values represent the highest Precision and AUC (between the classifiers processed by CF and datasets authors).

Dataset	**Ref**	**Approach**	**Model**	**Measure**	**Value %**
14.Phishing websites	[174]	CF	MLP	Precision	98.00
		Authors	MLP		94.00
15.Chronic Kidney	[175]	CF	MLP	AUC	99.75
		Authors	MLP		99.33
16.Physical activity	[159]	CF	Bagging	Precision	99.60
		Authors	Bagging		96.60
17.Companies bankruptcy 1	[176]	CF	C4.5	AUC	77.00
		Authors	C4.5		71.70
18.Companies bankruptcy 2	[176]	CF	C4.5	AUC	79.30
		Authors	C4.5		65.30
19.Companies bankruptcy 3	[176]	CF	C4.5	AUC	80.50
		Authors	C4.5		70.10
20.Companies bankruptcy 4	[176]	CF	C4.5	AUC	80.20
		Authors	C4.5		69.10
21.Companies bankruptcy 5	[176]	CF	C4.5	AUC	83.40
		Authors	C4.5		76.10
22.Bank telemarketing	[177]	CF	MLP	AUC	92.60
		Authors	MLP		92.90
23.Chemi. biodegradability	[178]	CF	Boosting	AUC	95.50
		Authors	Boosting		92.10
24.Risk cervical cancer	[179, 180]	CF	C4.5	AUC	93.20
		Authors	C4.5		53.30
25.Seismic hazard predic.	[181]	CF	CART	Precision	93.70
		Authors	CART		87.00
26.Vertebral column diagn.	[182, 183]	CF	MLP	Precision	85.50
		Authors	MLP		83.00
27.Vertebral column injury	[182, 183]	CF	SVM	Precision	88.20
		Authors	SVM		82.10
28.Voice rehabilitation	[184]	CF	SVM	Precision	88.10
		Authors	SVM		74.80

The classifiers were built with the dataset processed by the authors and the dataset cleaned by the conceptual framework (CF). The performance measures of the classifiers corresponding to the *Precision* and *Area Under Curve* (AUC). The UCI datasets were tested with other classifiers, the results of these classifiers are presented in Appendix A.3.1.

The values underlined in the Table 3.7 correspond to the highest *Precision* and *AUC*. Once cleaned the datasets by CF, 85.71% of the models achieve the highest *Precision* and *AUC* than models proposed by datasets authors. The remaining 14.81% correspond to the models of the dataset authors: “1. Anuran families calls”, “2. Anuran species calls”, “22. Bank telemarketing” and “12. Phishing detection”. In case of “1. Anuran families calls” and “2. Anuran species calls” the precisions difference of the *MLP* generated by authors respect to *MLP* built with datasets processed by CF are 1.4% and 0.1%, while the precisions difference of “22. Bank telemarketing” is 0.3%. For “12. Phishing detection”, the *Area Under Curve* generated by *CART* model of the dataset authors covers 6.2% more than *CART* model of CF.

In terms of *Precision* measure, our approach obtained more than 9% of *Precision* respect to classifiers processed by datasets authors: “3. Autism in adolescent”, “4. Autism in adult”, “5. Autism in child”, “10. Ozone level 1 hour”, “11. Ozone level 8 hours” and “28. Voice rehabilitation” as show Figure 3.13. In general, the *Average Precision* of the classifiers processed by Conceptual Framework (CF) reached 94.9% compared with 83.6% of *Average Precision* of the classifiers processed by datasets authors.

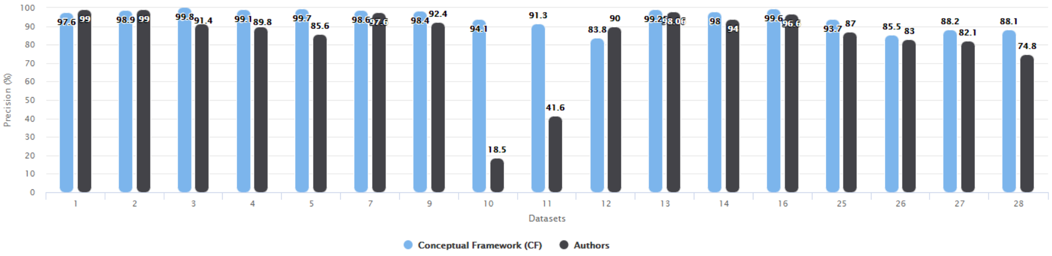


Figure 3.13: Comparison of the precision of classifiers generated from datasets created by the Conceptual Framework (CF) and the classifiers generated from the original datasets proposed by the authors and published in the UCI repository.

In case of *AUC* measure, the classifiers generated from dataset cleaned by CF reached more than 10% of *Area Under Curve* than the classifiers of the dataset authors of: “8. Default of credit card”, “18. Companies bankruptcy 2”, “19.

Companies bankruptcy 3", "20. Companies bankruptcy 4" and "24. Risk cervical cancer" as depict Figure 3.14. In summary, the *Average AUC* of the classifiers generated from dataset cleaned by CF achieved 87.02% compared with 76.83% of *Average AUC* of the classifiers processed by datasets authors.

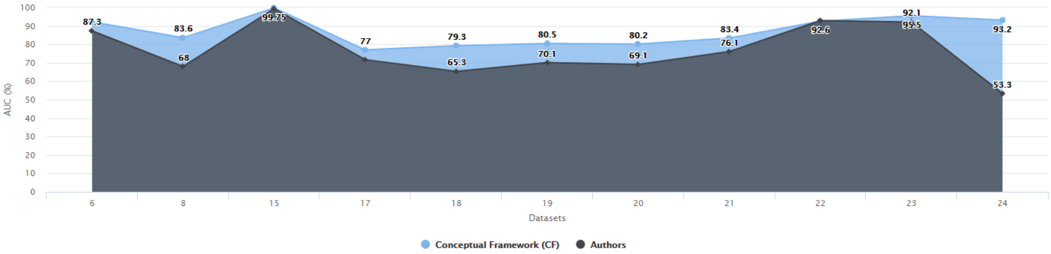


Figure 3.14: Comparison of the AUC of classifiers generated from datasets created by the Conceptual Framework (CF) and the classifiers generated from the original datasets proposed by the authors and published in the UCI repository.

Although we compared the results obtained by the classifiers trained with the cleaned datasets by CF and authors of UCI datasets, the comparison process is not enough due:

- The dataset authors omit details about the process of data preparation as the creation and modification of attributes from original ones, model validation technique (cross-validation, test set, etc.), or experimental configuration of the models. We followed the same experimental process with the available information (raw datasets and information of the datasets as forums and publications).
- In addition, the original dataset and the dataset cleaned by CF are different. The datasets differ mainly in the number of instances and attributes because we used several data cleaning tasks through CF.

With the aim to build a fair comparison process, we proposed a mini-challenge for the evaluation of the datasets (cleaned by the CF and original). In the next subsection, we present the mini-challenges for classification datasets.

3.5.1.3 Classification mini-challenges

The challenges address problems about knowledge discovery in data defined by a set of experts. The challenges offer rewards to the winner. An example of challenge is presented by KDD Cup which is the annual Data Mining and Knowledge Discovery competition organized by ACM Special Interest Group on Knowledge

Discovery and Data Mining [185]. In our case, we organized an experimental mini-challenge with the aim to demonstrate the capabilities of CF compared with the original dataset. The mini-challenges consider the following steps:

1. The original dataset is split in 80% for training and 20% for testing. To guarantee the same percentage of samples for each class label as the complete set, we selected the training and test set based on a stratified sampling [186].
2. The training set is cleaned with the CF.
3. We built a set of classifiers with the original training set and the training set cleaned by the CF. The algorithms used to build the classifiers correspond to algorithms used by authors of dataset published in UCI repository.
4. A significance test is applied to classifiers generated from the original training set and the classifiers built with training set cleaned by CF.
5. The best classifiers statistically significant are selected.
6. The best classifiers statistically significant are evaluated through test set.

The mini-challenge was carried out for three kinds of datasets:

- The original dataset with the highest similarity with respect dataset cleaned by CF.
- The original dataset with medium similarity with respect dataset cleaned by CF.
- The original dataset with the lowest similarity with respect dataset cleaned by CF.

We computed the similarity degree between original dataset and dataset processed by CF from twelve meta-features: *instances, attributes, data dimensionality, missing values ratio, duplicate instances ratio, mean absolute linear correlation, equivalent number of features, mean absolute skewness, mean absolute kurtosis, mean attribute entropy, mean mutual information, noise-signal ratio*. To select the meta-features, we reviewed several works which are analyzed in Subsection 5.1.1.

We computed local similarity for each meta-feature based on similarity measures as Euclidean, Arithmetic, and Canberra. Subsequently, we computed the global similarity which is given by the average of the local similarities. The mechanism to compute the similarity is presented in Subsection 5.2.2. Figure 3.15 shows the global similarity between original dataset and dataset cleaned by CF.

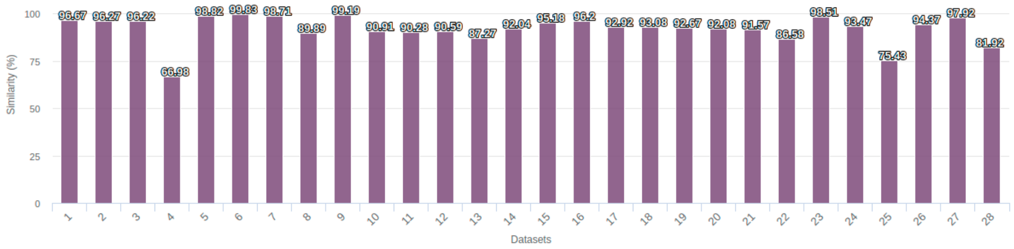


Figure 3.15: Similarity between dataset authors and dataset cleaned by CF - Classification tasks

Based on global similarity between dataset authors and dataset cleaned by CF, we selected the datasets with highest, median and lowest similarity degree, and we applied the mini-challenges where the performance measure of the classifiers corresponding to the *Precision*:

- Dataset 9: Human activity recognition. Datasets (authors and cleaned by CF) with the highest similarity degree.
- Dataset 28: Voice rehabilitation. Datasets (authors and cleaned by CF) with medium similarity degree
- Dataset 4: Autism in adult. Datasets (authors and cleaned by CF) with the lowest similarity degree.

Similarly, we applied the mini-challenges in the datasets with highest, median and lowest similarity degree where the performance measure of the classifiers corresponding to the *AUC*:

- Dataset 6: Breast tissue detection. Datasets (authors and cleaned by CF) with the highest similarity degree
- Dataset 18: Companies bankruptcy 2. Datasets (authors and cleaned by CF) with medium similarity degree
- Dataset 22: Bank telemarketing. Datasets (authors and cleaned by CF) with the lowest similarity degree.

The six mini-challenges are presented below.

Dataset 9: Human activity recognition

This dataset contains the highest global similarity presented in Figure 3.13 for Precision measure. The aim of this dataset is centered in Human Activity Recognition (HAR) using smartphones [170]. The raw dataset of the authors contains 4252 instances while the training set defined for the mini-challenge contains 3402 instances. Table 3.8 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 98.16%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.89%).

Table 3.8: Dataset 9: Human activity recognition. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	561	561	100	Canberra
Instances	4252	3402	88.895	Canberra
Data dimensionality	0.132	0.165	88.895	Canberra
Mean abs. Skewness	2.090	2.002	97.855	Canberra
Mean abs. Kurtosis	0.004	0.004	97.855	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.007	0	99.3	Euclidean
Class Entropy	0.995	0.992	99.709	Euclidean
		Similarity	98.167 %	

The original training set has a global similarity of 99.83 % respect to the training set cleaned by the CF as show Table 3.9. These datasets have a high global similarity due to CF applied just one data cleaning task:

- Dimensionality reduction: this data cleaning task discarded six attributes. Thus, the datasets have 93.46% of similarity between attributes.

- As a consequence of dimensionality reduction, the meta-features mean absolute skewness, mean absolute kurtosis and data dimensionality changed slightly. Thus, mean absolute skewness presents 99.53% of similarity , mean absolute kurtosis 98.99% and data dimensionality 99.46%.

Table 3.9: Dataset 9: Human activity recognition. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	561	555	99.462	Canberra
Instances	3402	3402	100	Canberra
Data dimensionality	0.165	0.163	99.462	Canberra
Mean abs. Skewness	2.002	2.021	99.531	Canberra
Mean abs. Kurtosis	0.004	0.004	98.993	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0	0	100	Euclidean
Class Entropy	0.992	0.992	100	Euclidean
		Similarity	99.830 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset “Human activity recognition” with the original training set and the training set cleaned by the CF. The authors of this dataset used one algorithm: Support Vector Machine (SVM). Finally, we validated the SVM classifiers with test set defined for the mini-challenge. The test set contains 106 instances. Thus, the SVM built with training set cleaned by the CF achieves the highest Accuracy 79.34%, compared with 71.69% Accuracy of the SVM built with the original training set.

Dataset 28: Voice rehabilitation

This dataset addresses the voice rehabilitation treatment [184]. The global similarity of this dataset is close to the average between the highest and lowest global similarity presented in Figure 3.13 for Precision measure. The raw dataset of the authors contains 126 instances while the training set defined for the mini-challenge contains 101 instances. Table 3.10 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 98.07%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.98%).

Table 3.10: Dataset 28: Voice rehabilitation. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	310	310	100	Canberra
Instances	126	101	88.987	Canberra
Data dimensionality	2.460	3.069	88.987	Canberra
Mean abs. Skewness	3.584	3.465	98.312	Canberra
Mean abs. Kurtosis	0.012	0.011	98.312	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0	0	100	Euclidean
Class Entropy	0.918	0.892	97.327	Euclidean
Imbalance Ratio	2	1.971	99.259	Canberra
		Similarity	98.079 %	

The original training set has a global similarity of 89.84 % respect to the training set cleaned by the CF as show Table 3.11. The main differences between original training set and training set cleaned by CF are caused by application of the data cleaning tasks:

- Classes balancing: 34 instances were generated from minority class. This data cleaning task reduces the similarity for meta-features instances (85.59%), class entropy (89.16%) and imbalance ratio (67.32%).

- Dimensionality reduction: this data cleaning task reduced considerably the dimensionality of the dataset with the elimination of 250 attributes. Thus, the datasets have 32.432% of similarity between attributes and 85.59% of similarity for data dimensionality.

Table 3.11: Dataset 28: Voice rehabilitation. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	310	60	32.432	Canberra
Instances	101	135	85.593	Canberra
Data dimensionality	3.069	2.296	85.593	Canberra
Mean abs. Skewness	3.465	3.926	93.764	Canberra
Mean abs. Kurtosis	0.011	0.013	93.764	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0	0	100	Euclidean
Class Entropy	0.892	1.000	89.160	Euclidean
Imbalance Ratio	1.971	1.000	67.327	Canberra
		Similarity	89.842 %	

Finally, we trained the same algorithms proposed by authors of the dataset “Voice rehabilitation” with the original training set and the training set cleaned by the CF. The authors of this dataset used Support Vector Machine (SVM). The classifiers were validated with test set defined for the mini-challenge. The test set contains 25 instances. Thus, the SVM built with training set cleaned by the CF achieves the highest Accuracy 100%, compared with 84% Accuracy of the SVM built with the original training set.

Dataset 4: Autism in adult

This dataset contains the lowest global similarity presented in Figure 3.13 for Precision measure. This dataset describes the detection of Autism Spectrum Disorder (ASD) in adults [166]. The raw dataset of the authors contains 704 instances while the training set defined for the mini-challenge contains 563 instances. Table 3.12 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 97.96%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.71%).

Table 3.12: Dataset 4: Autism in adult. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	20	20	100	Canberra
Instances	704	563	88.871	Canberra
Data dimensionality	0.028	0.036	88.871	Canberra
Mean abs. Skewness	1.570	1.532	98.802	Canberra
Mean abs. Kurtosis	0.131	0.128	98.802	Arithmetic
Mean attribute entropy	0.515	0.518	99.704	Euclidean
Mean mutual information	0.028	0.027	99.586	Arithmetic
Mean abs. linear correlation	0.175	0.194	98.168	Euclidean
Equivalent num. of features	30.283	29.944	99.438	Canberra
Noise-signal ratio	17.594	17.857	99.260	Canberra
Missing values ratio	0.013	0.014	99.900	Euclidean
Duplicate instances ratio	0.007	0.004	99.700	Euclidean
Class Entropy	0.839	0.823	98.377	Euclidean
Imbalance Ratio	2	2	100	Canberra
		Similarity	97.965 %	

The original training set has a global similarity of 88.60 % respect to the training set cleaned by the CF as show Table 3.13. The low global similarity between these training sets is caused because the CF modified the original training set to apply the data cleaning tasks:

- Imputation: 1.4% of training set values were imputed (98.60% of local similarity in missing values ratio).
- Classes balancing: 146 instances were generated from minority class (88.87% of similarity in number of instances, 84.70% of similarity in class entropy and 66.66% in imbalance ratio).
- Remove duplicate instances: 0.4 % of duplicate instances of the training set were removed (99.60% of local similarity in duplicate instances ratio).
- Dimensionality reduction: the CF detected one redundant attribute and this attribute was discarded (97.43% of similarity between attributes).

Table 3.13: Dataset 4: Autism in adult. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	20	19	97.436	Canberra
Instances	563	709	88.871	Canberra
Data dimensionality	0.036	0.027	86.346	Canberra
Mean abs. skewness	1.532	1.489	98.559	Canberra
Mean abs. kurtosis	0.128	0.124	98.559	Arithmetic
Mean attribute entropy	0.518	0.508	98.988	Euclidean
Mean mutual information	0.027	0.051	69.795	Arithmetic
Mean abs. linear correlation	0.194	0.227	96.673	Euclidean
Equivalent num. of features	29.944	19.035	77.728	Canberra
Noise-signal ratio	17.857	8.911	66.578	Canberra
Missing values ratio	0.014	0.000	98.600	Euclidean
Duplicate instances ratio	0.004	0.000	99.600	Euclidean
Class Entropy	0.823	0.976	84.700	Euclidean
Imbalance Ratio	2.000	1.000	66.667	Canberra
		Similarity	88.607 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset "Autism in adult" with the original training set and the training set cleaned by the CF. The authors of this dataset used C4.5, Reduced Error Pruning (REP) Tree and Random Forest (RF). With aim to select the classifiers statistically better, we applied paired sample (t-test) [187] with $\rho = 0.5$. Tables 3.14 and 3.15 present the Accuracy for the classifiers built with the original training set and the training set cleaned by CF.

Table 3.14: Dataset 4: Autism in adult (Training set). Accuracy for C4.5, REP Tree and RF. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	Accuracy	C4.5	REP Tree	RF
C4.5	100%		(.)	(.)
REP Tree	100%	(.)		(.)
RF	100%	(.)	(.)	

Table 3.15: Dataset 4: Autism in adult (Training set cleaned by CF). Accuracy for C4.5, REP Tree and RF. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	Accuracy	C4.5	REP Tree	RF
C4.5	100%		(.)	(.)
REP Tree	100%	(.)		(.)
RF	100%	(.)	(.)	

The classifiers (C4.5, REP Tree and RF) presented in Tables 3.14 and 3.15 do not present statistically significant differences. Thus, we validated all classifiers through test set defined for the mini-challenge. The test set contains 141 instances. All classifiers (C4.5, REP Tree and RF) reached 100% of Accuracy for both training sets.

Dataset 6: Breast tissue detection

This dataset contains the highest global similarity presented in Figure 3.14 for AUC measure. The dataset contains electrical impedance measurements of tissue samples from the breast [167]. The raw dataset of the authors contains 106 instances while the training set defined for the mini-challenge contains 84 instances. Table 3.16 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 98.31%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.42%).

Table 3.16: Dataset 6: Breast tissue detection. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	9	9	100	Canberra
Instances	106	84	88.421	Canberra
Data dimensionality	0.085	0.107	88.421	Canberra
Mean abs. Skewness	2.254	2.289	99.229	Canberra
Mean abs. Kurtosis	0.250	0.254	99.229	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.009	0.012	99.7	Euclidean
Class Entropy	0.992	0.989	99.655	Euclidean
		Similarity	98.310 %	

The original training set has a global similarity of 99.77 % respect to the training set cleaned by the CF as show Table 3.17. These datasets have a high global similarity due to CF applied just one data cleaning task:

- Remove duplicate instances: this data cleaning task removed 1.2% of duplicate instances. Thus, the datasets have 99.40% of similarity between attributes.
- As a consequence of remove duplicate instances, the meta-features mean absolute skewness and kurtosis, instances and data dimensionality changed slightly; 99.64% of similarity for meta-features mean absolute skewness and kurtosis, 99.40% of similarity for meta-features data dimensionality and instances.

Table 3.17: Dataset 6: Breast tissue detection. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	9	9	100	Canberra
Instances	84	83	99.401	Canberra
Data dimensionality	0.107	0.108	99.401	Canberra
Mean abs. Skewness	2.289	2.272	99.643	Canberra
Mean abs. Kurtosis	0.254	0.252	99.643	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.012	0	98.8	Euclidean
Class Entropy	0.989	0.987	99.797	Euclidean
		Similarity	99.779 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset "Breast tissue detection" with the original training set and the training set cleaned by the CF. The authors of this dataset used Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA). With aim to select the classifiers statistically better for each training set, we applied paired sample (t-test) [187] with $\rho = 0.5$. Tables 3.18 and 3.19 present the AUC for the classifiers built with the original training set and the training set cleaned by CF.

Table 3.18: Dataset 6: Breast tissue detection (Training set). AUC measure for SVM and LDA. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	AUC	SVM	LDA
SVM	76%		(+)
LDA	74%	(-)	

Table 3.19: Dataset 6: Breast tissue detection (Training set cleaned by CF). AUC measure for SVM and LDA. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	AUC	SVM	LDA
SVM	77%		(+)
LDA	76%	(-)	

In both training sets, Support Vector Machine is significantly better than remain of classifiers. The SVM built with original training set achieved 76% AUC, while the SVM of the training set cleaned by CF reached 77% AUC.

Finally, SVM classifiers were validated with test set defined for the mini-challenge. The test set contains 22 instances. The SVM of the training set cleaned by CF reached the highest AUC 85.8% compared with 84.2% AUC achieved by SVM built with the original training set.

Dataset 18: Companies bankruptcy 2

The dataset contains information about bankruptcy Polish companies [176]. The global similarity of this dataset represents the average between the highest and lowest global similarity presented in Figure 3.14 for AUC measure. The raw dataset of the authors contains 10173 instances while the training set defined for the mini-challenge contains 8138 instances. Table 3.20 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 96.58%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.88%).

Table 3.20: Dataset 18: Companies bankruptcy 2. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	64	64	100	Canberra
Instances	10173	8138	88.886	Canberra
Data dimensionality	0.006	0.008	88.886	Canberra
Mean abs. Skewness	76.213	66.998	93.565	Canberra
Mean abs. Kurtosis	1.191	1.047	93.565	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0.018	0.018	100	Euclidean
Duplicate instances ratio	0.009	0	99.1	Euclidean
Class Entropy	0.239	0.283	95.638	Euclidean
Imbalance Ratio	24	19	88.372	Canberra
		Similarity	96.581 %	

The original training set has a global similarity of 93.26 % respect to the training set cleaned by the CF as show Table 3.21. The main differences between the original training set and training set cleaned by CF are caused by application of the data cleaning tasks:

- Imputation: 1.8% of training set values were imputed (98.60% of local similarity in Missing values ratio).
- Classes balancing: 566 instances were generated from minority class. This data cleaning task reduces the similarity for meta-features instances (96.63%), Class Entropy (76.86%) and Imbalance Ratio (53.84%).
- Remove duplicate instances: 0.4 % of duplicate instances of the training set were removed (99.70% of local similarity in Duplicate instances ratio).
- Dimensionality reduction: this data cleaning task reduced the dimensionality of the dataset with the elimination of 9 attributes. Thus, the datasets have 92.437% of similarity between attributes and 89.104% of similarity for data dimensionality.

Table 3.21: Dataset 18: Companies bankruptcy 2. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	64	55	92.437	Canberra
Instances	8138	8704	96.639	Canberra
Data dimensionality	0.008	0.006	89.104	Canberra
Mean abs. Skewness	66.998	66.423	99.569	Canberra
Mean abs. Kurtosis	1.047	1.208	92.865	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean mutual information	0	0	100	Arithmetic
Mean abs. linear correlation	0	0	100	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0.018	0	98.2	Euclidean
Duplicate instances ratio	0.007	0.000	99.70	Euclidean
Class Entropy	0.283	0.514	76.863	Euclidean
Imbalance Ratio	19	7	53.846	Canberra
		Similarity	93.268 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset “Companies bankruptcy 2” with the original training set and the training set cleaned by the CF. The authors of this dataset used C4.5 Decision Tree, Multi Layer Perceptron (MLP) and Support Vector Machine (SVM). With aim to select the classifiers statistically better for each training set, we applied paired sample (t-test) [187] with $\rho = 0.5$. Tables 3.22 and 3.23 present the AUC for the classifiers built with the original training set and the training set cleaned by CF.

Table 3.22: Dataset 18: Companies bankruptcy 2 (Training set). AUC measure for C4.5, MLP and SVM. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	AUC	C4.5	MLP	SMV
C4.5	65%		(-)	(+)
MLP	75%	(+)		(+)
SMV	50%	(-)	(-)	

Table 3.23: Dataset 18: Companies bankruptcy 2 (Training set cleaned by CF). AUC measure for C4.5, MLP and SVM. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	AUC	C4.5	MLP	SMV
C4.5	71%		(+)	(+)
MLP	52%	(-)		(.)
SMV	50%	(-)	(.)	

For original training set presented in Table 3.22, Multi Layer Perceptron (75% AUC) is significantly better than C4.5 decision tree and Support Vector Machine. In case of training set cleaned by CF which is presented in 3.23, C4.5 decision tree (71% AUC) is significantly better than remain of classifiers.

Thus, we validated the classifiers significantly better of the training sets (MLP of the original training set and C4.5 of training set cleaned by CF) through test set defined for the mini-challenge. The test set contains 2035 instances. MLP achieved the highest Accuracy (99.26%) compared with Accuracy reached by C4.5 (96.26%).

Dataset 22: Bank telemarketing

This dataset contains the lowest global similarity presented in Figure 3.13 for AUC measure. This dataset contains information about marketing campaigns through phone calls of a portuguese banking institution [177]. The raw dataset of the authors contains 45211 instances while the training set defined for the mini-challenge contains 36169 instances. Table 3.24 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 95.42%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.88%).

Table 3.24: Dataset 22: Bank telemarketing. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	16	16	100	Canberra
Instances	45211	36169	88.889	Canberra
Data dimensionality	0.0003	0.0004	88.889	Canberra
Mean abs. Skewness	8.806	8.187	96.359	Canberra
Mean abs. Kurtosis	1.258	1.170	96.359	Arithmetic
Mean attribute entropy	0.697	0.693	99.576	Euclidean
Mean mutual information	0.010	0.009	93.915	Arithmetic
Mean abs. linear correlation	0.160	0.140	97.965	Euclidean
Equivalent num. of features	50.428	63.296	88.685	Canberra
Noise-signal ratio	66.548	74.838	94.137	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0	0	100	Euclidean
Class Entropy	0.521	0.579	94.212	Euclidean
Imbalance Ratio	7	6	92.308	Canberra
		Similarity	95.420 %	

The original training set has a global similarity of 82.35 % respect to the training set cleaned by the CF as show Table 3.25. The low global similarity between these training sets is due the data cleaning process made by CF:

- Classes balancing: 7477 instances were generated from minority class (90.632% of similarity in the number of instances, 71.56% of similarity in Class Entropy and 50% in Imbalance Ratio).
- Dimensionality reduction: the CF detected two redundant attributes and there were discarded (93.33% of similarity between attributes).

Table 3.25: Dataset 22: Bank telemarketing. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	16	14	93.333	Canberra
Instances	36169	43646	90.632	Canberra
Data dimensionality	0.0003	0.0004	84.065	Canberra
Mean abs. Skewness	8.187	7.443	95.238	Canberra
Mean abs. Kurtosis	1.170	1.063	95.238	Arithmetic
Mean attribute entropy	0.693	0.638	94.458	Euclidean
Mean mutual information	0.009	0.025	53.461	Arithmetic
Mean abs. linear correlation	0.140	0.265	87.542	Euclidean
Equivalent num. of features	63.296	34.442	70.479	Canberra
Noise-signal ratio	74.838	24.456	49.259	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0	0	100	Euclidean
Class Entropy	0.579	0.863	71.565	Euclidean
Imbalance Ratio	6	2	50	Canberra
		Similarity	82.351 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset "Bank telemarketing" with the original training set and the training set cleaned by the CF. The authors of this dataset used C4.5 Decision Tree, Support Vector Machine (SVM) and Multi Layer Perceptron (MLP). With aim to select the classifiers statistically better for each training set, we applied paired sample (t-test) [187] with $\rho = 0.5$. Tables 3.26 and 3.27 present the AUC for the classifiers built with the original training set and the training set cleaned by CF.

Table 3.26: Dataset 22: Bank telemarketing (Training set). AUC measure for C4.5, SVM and MLP. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	AUC	C4.5	SVM	MLP
C4.5	82%		(+)	(-)
SVM	59%	(-)		(-)
MLP	87%	(+)	(+)	

Table 3.27: Dataset 22: Bank telemarketing (Training set cleaned by CF). AUC measure for C4.5, SVM and MLP. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	AUC	C4.5	SVM	MLP
C4.5	90%		(+)	(-)
SVM	80%	(-)		(-)
MLP	93%	(+)	(+)	

For training sets, Multi Layer Perceptron classifiers are significantly better than C4.5 decision tree and Support Vector Machine. The MLP built with original training set achieved 87% AUC, while the MLP of the training set cleaned by CF reached 93% AUC.

Thus, we validated the classifiers significantly better of the training sets (Multi Layer Perceptron classifiers) through test set defined for the mini-challenge. The test set contains 134 instances. MLP of the original training set classified 129 instances correctly (Accuracy 96.26%), while MLP of the training set cleaned by CF classified 128 instances correctly (Accuracy 95.52%).

In summary, 4/6 classification mini-challenges, the classifiers generated by the datasets cleaned by CF achieved the highest *Accuracy* and *AUC*.

3.5.2 Regression tasks

3.5.2.1 Test of conceptual framework: comments prediction in Facebook

The dataset for regression tasks was proposed in [160], which is oriented to the prediction of comments in a Facebook post. The dataset is composed of a data test with 10.120 instances and five training sets (Variant 1 - 5; these training sets were cleaned by CF) as shown Table 3.28.

Table 3.28: Number of instances of dataset for prediction of comments in Facebook posts

Data training	**Instances**
Variant 1	40.949
Variant 2	81.312
Variant 3	121.098
Variant 4	160.424
Variant 5	199.030

The dataset contains 53 attributes: 4 page features (page likes, page category, etc.), 30 essential features (comment count in last 24 and 48 hrs, etc.), 14 Weekday features (binary variables related with the date of Facebook post), and 5 other basic features (length of document, time gap between selected base date/time, document published date/time, document promotion status and post share count).

- *Imputation*: after executing the first step of the conceptual framework, we conclude that the original dataset does not contain missing values. With the goal of testing the imputation step, we remove values randomly from the original dataset using *R* statistical software [188]. As a result of this operation, the dataset presents missing values in three attributes. Therefore, we test two imputation approaches: *Random forest imputation* [143, 189] and *Mean imputation* [190]. Table 3.29 presents the Mean Absolute Error of the imputation methods.

Table 3.29: Mean absolute error for imputation methods: Random forest and mean imputation. Training sets: Variant 1- 5

Dataset	Att. Index	Random Forest	Mean Imputation
Variant 1	6	0.011	97.749
	26	0.001	1.752
	44	0.017	0.237
Variant 2	15	0.009	287.652
	31	1.214	18.624
	8	5.86 E-04	35.803
Variant 3	22	0	26.665
	48	0.004	0.203
	3	0.003	6.546
Variant 4	13	3.6 E-05	126.782
	49	0.010	0.233
	17	2.47 E-13	4.896
Variant 5	12	3.09 E-13	7.172
	29	0.135	54.445
	52	0.006	0.223

Random Forest reaches low *MAE* in the imputations (*MAE* lowest: 0 in attribute 22 Variant 3, and *MAE* highest: 1.214 in attribute 31 of Variant 3). In contrast with *Mean Imputation*, the attributes 6,15,13,29 shown in Table 3.29 have a *MAE* greater than 54.445. This happens because the imputation values were added on the center of the sample, diminishing the importance of values on the tails. Thus, *Random Forest* was the algorithm used for impute the missing values. Figure 3.16 presents the imputed (red dotted line) and original values (black dotted line) for attribute 6 (comments average in last 24 hours of the training set - Variant 1).

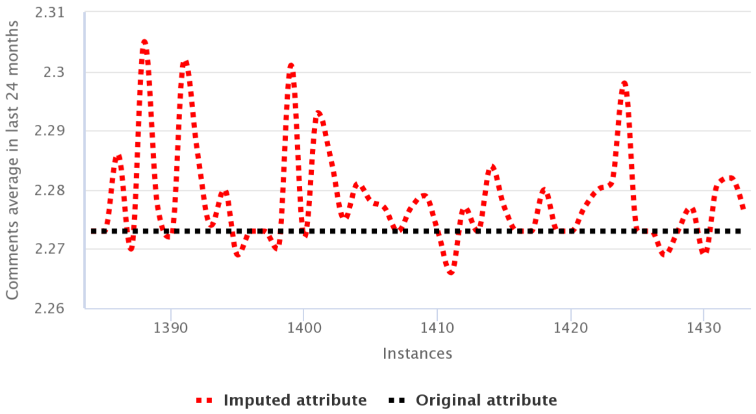


Figure 3.16: Data training - Variant 1: imputed values for Attribute 6

In Figure 3.16 is observed the imputed values are around 2.225 - 2.305, while the original values are 2.273. Thus the imputation obtained by *Random forest* reaches a *Mean Absolute Error* 0.01.
Other imputation for the attribute 31: comments in last 24 hours of the training set - variant 2 is shown in Figure 3.17.

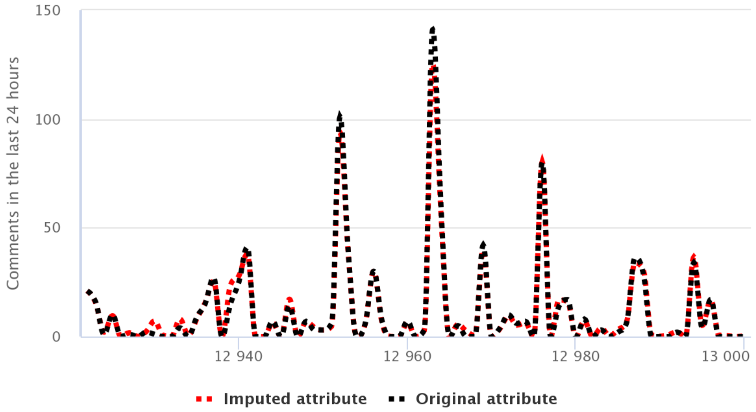


Figure 3.17: Data training - Variant 2: imputed values for Attribute 31

In this case the imputation method obtain a mean absolute error of 1.21.

- *Outliers detection*: once imputed values, according to the Conceptual Framework presented in Figure 3.8, we applied the outliers detection task with the aim to find abnormal behavioral in the instances or erroneous imputations. In this case, we propose the use of outliers detection based on distance (*Local Outlier Factor*) [144] and clustering (*Density-Based Spatial Clustering* *of Applications with Noise*) [191, 192] approaches. Table 3.30 shows the candidate outliers detected by *LOF* and *DBSCAN*.

Table 3.30: Outliers detected by LOF and DBSCAN

Data training	**LOF**	**DBSCAN**
Variant 1	7	134
Variant 2	2	113
Variant 3	6	97
Variant 4	11	179
Variant 5	13	219

The clusters of outliers created by *DBSCAN* reach among 97 and 219 instances (Table 3.30), however 97.35% of the instances considered outliers are false positives. In case of *Local Outlier Factor*, the instances with *LOF* scores greater than 4.134 were analyzed (among 2 and 13 instances depending of dataset as shown Table 3.30), obtaining that 100% of the candidate outliers are true positives.

From the foregoing *LOF* was the algorithm used for outliers detection. To verify the candidate outliers obtained by *LOF*, the first two principal components for each training sets were plotted. Figure 3.18 presents principal components PC1 and PC2 for training set - Variant 5; 99.99% of the information contained in the training set are retained by the first two components. The outliers are labeled with ”+” in red.

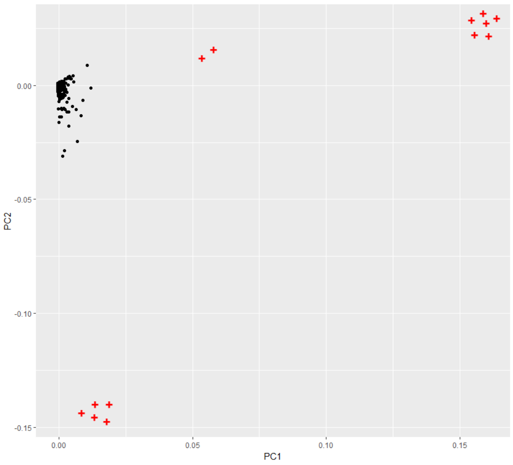


Figure 3.18: Outliers detected by LOF for training set - Variant 5. The outliers are represented by symbol + in red color. The black dots represent the instances.

The candidate outliers detected by *Local Outlier Factor* (Table 3.30) were removed which can be erroneous observations generated in the imputation task.

- *Remove duplicate instances*: we use the *Standard Duplicate Elimination* algorithm to detect duplicate instances [155]. They are removed by performing an external merge-sort and then scanning the sorted dataset. Similarly, we cluster and remove identical instances in a sequential scan of the sorted dataset [193]. Table 3.31 shows the number of duplicate instances for each training set (remove 312 duplicate instances).

Table 3.31: Duplicate instances for each training set

Data training	**Duplicate instances**
Variant 1	8
Variant 2	21
Variant 3	59
Variant 4	88
Variant 5	136

- *Dimensionality reduction*: considering that the datasets are large respect to low computational resources, we recommend using two methods of filter approach based on the absolute correlation. This methods are considered faster and they have low computational cost [194]. The absolute values of pair-wise correlations are considered. If two attributes have a high correlation, the filter algorithm looks at the mean absolute correlation of each attribute and removes the variable with the largest mean absolute correlation [195]. *Chi-squared* [196] and *Information Gain* [197, 198, 199] were the methods used. Figures 3.19, 3.20, 3.21, 3.22 and 3.23 show the absolute correlation for each attribute reached by *Chi-squared* and *Information gain*. The filter methods obtained a similar absolute correlation for the attributes of all datasets. The attributes with an absolute correlation of 0.2 or lower were removed (index of attributes removed: 4,9,14,19,35,37-52. Appendix A.2 presents the description of the attributes).

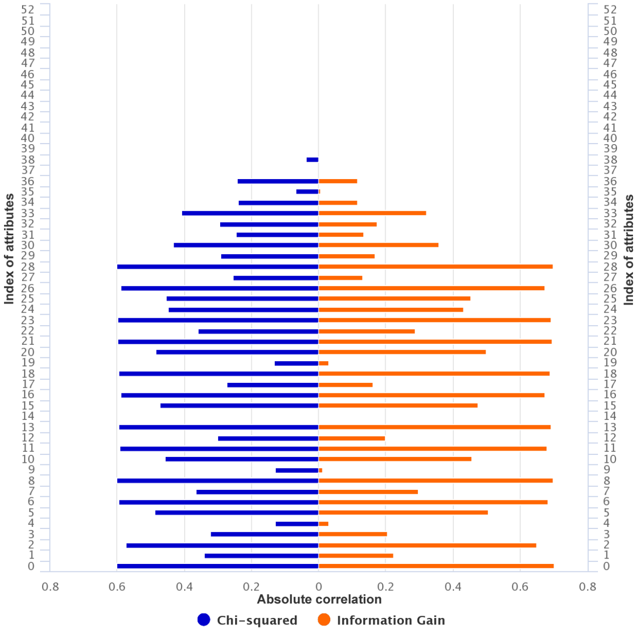


Figure 3.19: Absolute correlation obtained by Chi-squared and Information gain for Variant 1

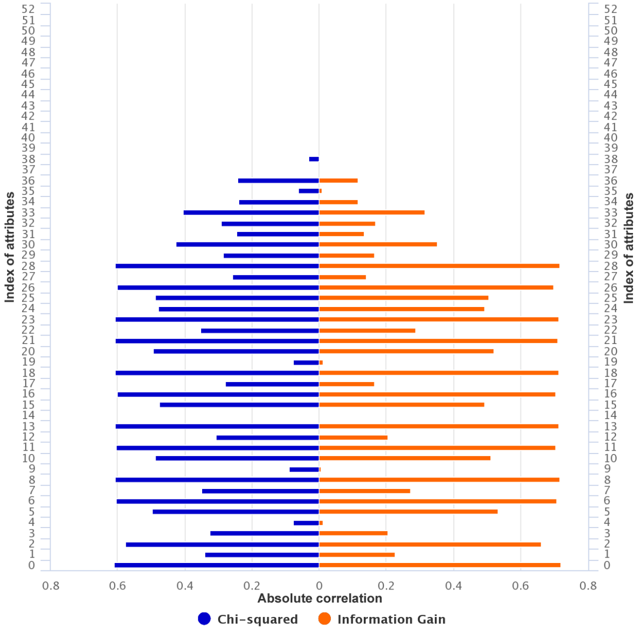


Figure 3.20: Absolute correlation obtained by Chi-squared and Information gain for Variant 2

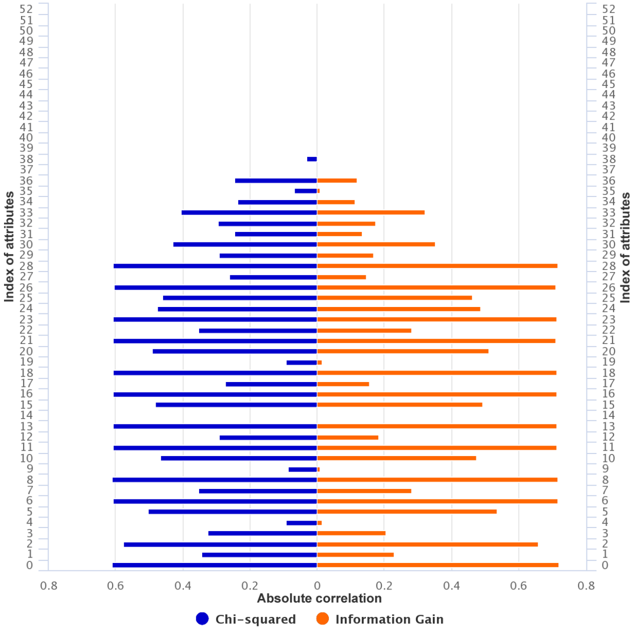


Figure 3.21: Absolute correlation obtained by Chi-squared and Information gain for Variant 3

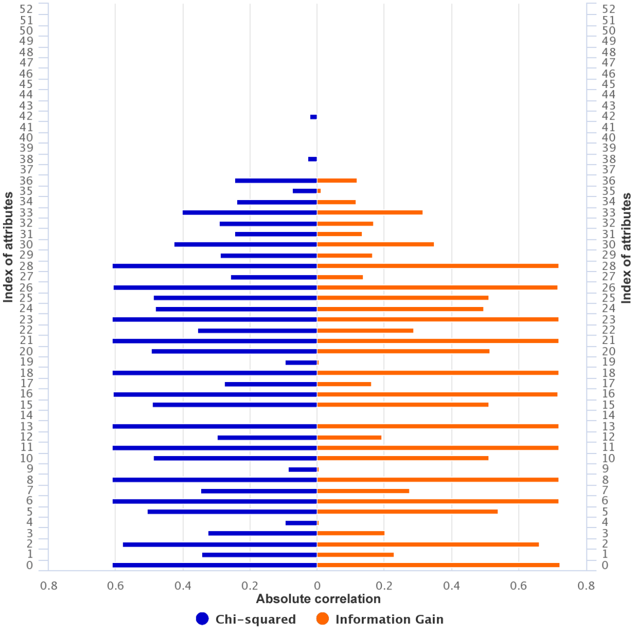


Figure 3.22: Absolute correlation obtained by Chi-squared and Information gain for Variant 4

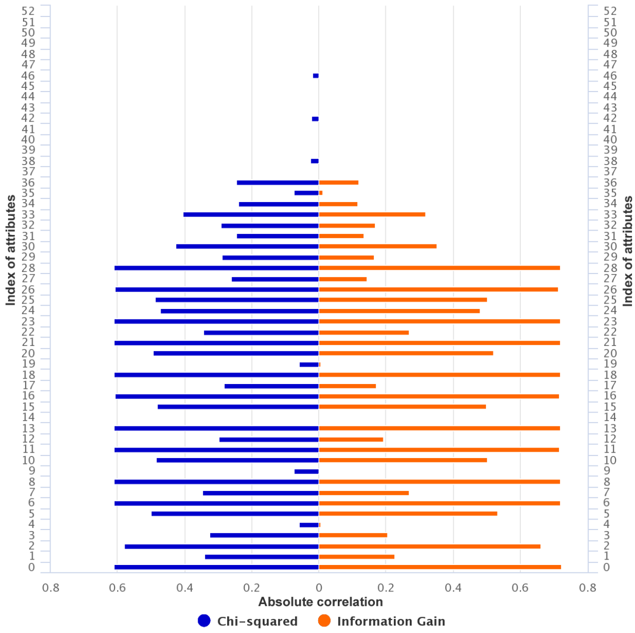


Figure 3.23: Absolute correlation obtained by Chi-squared and Information gain for Variant 5

- *Results*: with the aim of assessing of conceptual framework for regression task, we use the cleaned dataset by the conceptual framework for training the same regression models proposed by the authors of cFp dataset [160]. Then, we compare the results of *MAE* obtained by the two approaches. Authors of [160] used four regression algorithms of the Weka toolkit:

  - *Multi Layer Perceptron (MLP)*: this neural network was designed with two hidden layers; the first hidden layer contains 20 neurons while the second hidden layer 4 neurons. The learning rate is adjusted to 0.1 and momentum to 0.01.
  - *Radial Basis Function Network (RBF)*: the number of clusters was modified to 90.
  - In the models *REP* and *M5P Tree* were used the default parameters.

  The regression models were evaluated with a data test set of 10.120 instances. We do not use a statistical significance test due to the datasets (original and cleaned by CF) are different. The datasets differ mainly in the number of instances and attributes because we used several data cleaning tasks. Table 3.32 shows the MAE of the models generated by dataset cleaned with CF and the models proposed by the authors of cFp dataset

[160]; the underlined values represent the lowest *MAE* overall achieved by the models using CF and the authors proposal [160].

Table 3.32: MAE obtained by: Conceptual framework (CF) and [160]. The underlined values represent the lowest *MAE* overall achieved by the models.

Approach	**Model**	**Var 1**	**Var 2**	**Var 3**	**Var 4**	**Var 5**
CF	MLP	34.55	31.31	35.19	38.59	55.17
	RBF	31.09	31.85	30.12	29.81	29.69
	REP	29.28	30.22	28.41	27.89	29.33
	M5P	35.53	30.32	32.68	50.77	32.59
	Overall	32.61	30.92	31.60	36.76	34.19
[160]	MLP	38.24	40.72	36.40	51.49	44.93
	RBF	31.38	30.08	30.22	32.67	31.37
	REP	27.00	28.67	27.92	27.47	27.72
	M5P	30.15	36.90	32.33	35.69	116.98
	Overall	31.69	34.09	31.71	41.33	55.25

REP Tree was the model with lowest *MAE* for CF and authors proposal [160]. Whereas, *M5P tree* of [160] (training with Variant 5) was the model with highest *MAE*.

In overall, the regression models built with training sets Variant 2, 3, 4, 5 (cleaning by CF) achieved the lowest *MAE*. In case of Variant 1, the authors proposal [160] reaches a MAE lowest with a difference of 0.92 *MAE* overall respect to CF.

3.5.2.2 Comparative study

Similarly to results of the classification tasks, the CF was tested with 20 datasets coming from UCI Repository of Machine Learning Databases [158] for regression tasks. We used the same classifiers proposed by the dataset authors: Support Vector Regression (SVR), Linear Regression (LR), Random Forest (RF), M5P Decision Tree, and Multi Layer Perceptron (MLP). Table 3.33 presents two classifiers for each UCI dataset. The classifiers were built with the dataset processed by the authors and the dataset cleaned by the conceptual framework (CF). The classifiers were evaluated through Mean Absolute Error (MAE). In addition, the UCI datasets were tested with other classifiers, the results of these classifiers are presented in Appendix A.3.2.

Table 3.33: Mean absolute errors of the models processed by conceptual framework (CF) and datasets authors of UCI repository

Dataset	Ref	Approach	Model	MAE
1.Airfoil Self Noise	[200]	CF	LR	13.78
		Authors	LR	19.21
2.Beijing PM 2.5 pollution	[201]	CF	LR	2.53
		Authors	LR	6.55
3.Comments prediction in FB – 1	[160]	CF	MLP	34.55
		Authors	MLP	38.24
4.Comments prediction in FB – 2	[160]	CF	MLP	31.31
		Authors	MLP	40.72
5.Comments prediction in FB – 3	[160]	CF	RBF	30.12
		Authors	RBF	30.22
6.Comments prediction in FB – 4	[160]	CF	RBF	29.81
		Authors	RBF	32.67
7.Comments prediction in FB – 5	[160]	CF	M5P	32.59
		Authors	M5P	116.98
8.Compressor decay	[202]	CF	SVR	0.005
		Authors	SVR	0.17
9.Turbine decay	[202]	CF	SVR	0.003
		Authors	SVR	0.001
10.Rental Bikes Hourly	[203]	CF	LR	1e-05
		Authors	LR	0.017
11.Air Pollution Benzene	[204, 205]	CF	MLP	8.33
		Authors	MLP	11.50
12.Rental Bikes Daily	[203]	CF	LR	5e-05
		Authors	LR	0.031
13.Energy use of appliances	[206]	CF	RF	12.03
		Authors	RF	11.97
14.Posts in Facebook pages	[207]	CF	SVR	25.26
		Authors	SVR	26.9
15.Feedback Blogs Prediction	[208]	CF	M5P	5.70
		Authors	M5P	6.06

Table 3.33: Mean absolute errors of the models processed by conceptual framework (CF) and datasets authors of UCI repository

Dataset	Ref	Approach	Model	MAE
16.Forest Fires	[209]	CF	SVR	<u>4.60</u>
		Authors	SVR	12.71
17.I-Room temperature	[210]	CF	MLP	<u>0.47</u>
		Authors	MLP	1.13
18.II-Room temperature	[210]	CF	MLP	<u>0.34</u>
		Authors	MLP	0.88
19.I-Dinning room temperature	[210]	CF	MLP	<u>0.43</u>
		Authors	MLP	0.89
20.II-Dinning room temperature	[210]	CF	MLP	<u>0.32</u>
		Authors	MLP	0.78

The values underlined in Table 3.33 correspond to the *MAE* lowest. Once cleaned the regression datasets by our conceptual framework, 90% of the models reach *Mean Absolute Error* less than models proposed by datasets authors. For remaining 12.5% of the models, the authors proposal of the datasets: “9. Turbine decay” and “13. Energy uses of appliances” achieve lowest *MAE*. In case of “9. Turbine decay” dataset, the *MAE* difference of *SVR* models is 0.002 and 0.06 for “13. Energy uses of appliances” dataset, using *RF* models.

In terms of *Mean Absolute Error (MAE)*, our approach obtained a lowest *MAE* (32.59) compared with *MAE* (116.98) obtained by the classifier of the dataset authors: “7.Comments prediction in FB – 5”. Similarly, for dataset: “4.Comments prediction in FB – 2”, CF reached *MAE* (31.31) lowest compared with classifier of the authors (40.72). In case of dataset: “16.Forest Fires”, CF reached *MAE* 4.6 respect to *MAE* 12.71 of the classifier processed by authors as show Figure 3.24. In general, the *Average MAE* of the classifiers generated from dataset cleaned by CF reached 11.60 compared with 17.88 of *Average MAE* of the classifiers created from datasets authors.

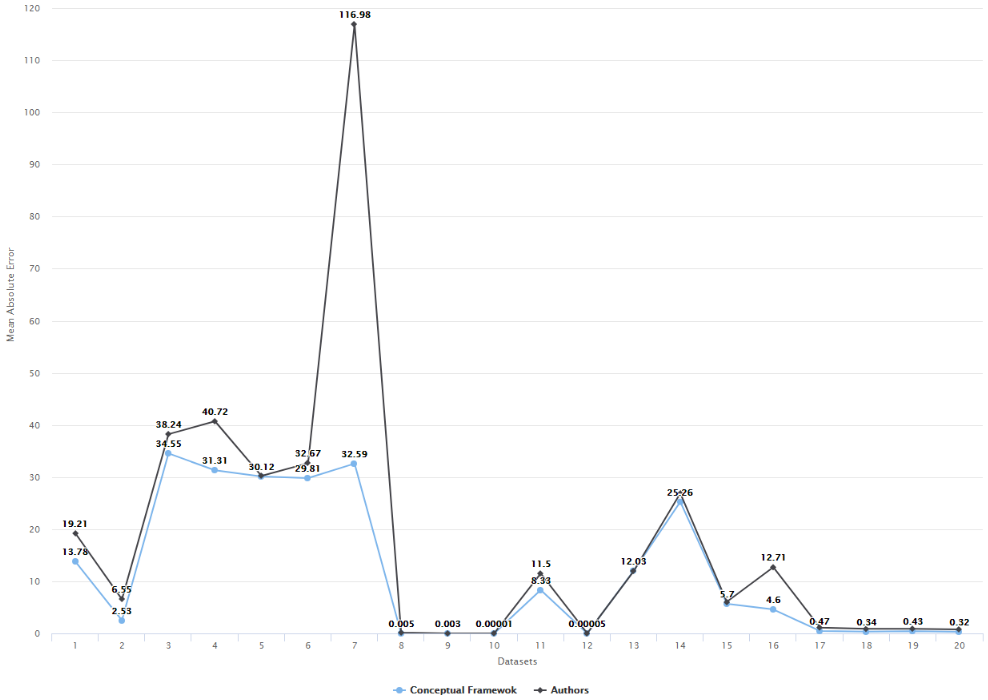


Figure 3.24: Mean absolute errors of the models processed by conceptual framework (CF) and datasets of authors of UCI repository

Similarly to classification datasets (Subsection 3.5.1.2), the results obtained by the regression models trained with the cleaned datasets by CF and authors of UCI datasets are not enough to evaluate the performance of the regression models due the dataset authors omit details about the process of data preparation as the creation and modification of attributes from original ones, model validation technique (cross-validation, test set, etc.), or experimental configuration of the models. We followed the same experimental process with the available information (raw datasets and information of the datasets as forums and publications). In addition, the original dataset and the dataset cleaned by CF are different. The datasets differ mainly in the number of instances and attributes because we used several data cleaning tasks through CF.

As classification mini-challenges presented in subsection 3.5.1.3, we propose mini-challenges for the evaluation of the regression datasets (cleaned by the CF and authors) which are presented in the next subsection.

3.5.2.3 Regression mini-challenges

Similar to classification mini-challenges, we organized an experimental mini-challenge for regression datasets with the aim to demonstrate the capabilities of CF compared with the original dataset. We computed the similarity degree between dataset authors and dataset processed by CF from twelve meta-features. Subsection 5.2.2 presents the mechanism to compute the similarity in detail. Figure 3.25 shows the global similarity between dataset authors and dataset cleaned by CF.

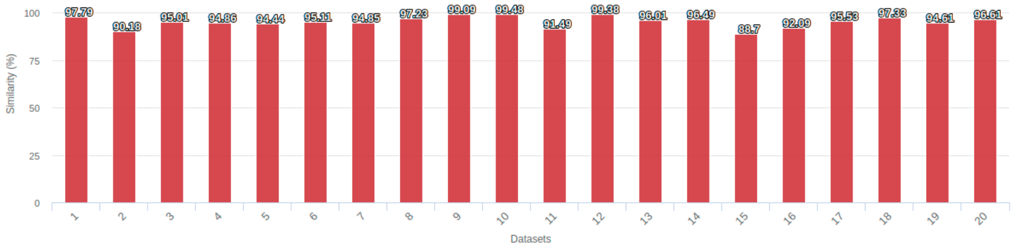


Figure 3.25: Similarity between dataset of authors and dataset cleaned by CF - Regression tasks

Based on global similarity between dataset authors and dataset cleaned by CF (Figure 3.25), we carried out the mini-challenges on datasets with highest, median and lowest similarity degree:

- Dataset 9: Human activity recognition. Datasets (authors and cleaned by CF) with the highest similarity degree.
- Dataset 28: Voice rehabilitation. Datasets (authors and cleaned by CF) with medium similarity degree
- Dataset 4: Autism in adult. Datasets (authors and cleaned by CF) with the lowest similarity degree.

The three mini-challenges are presented below.

Dataset 10: Rental Bikes Hourly (High)

This dataset contains the highest global similarity presented in Figure 3.25 for MAE measure. The dataset contains the hourly count of rental bikes between years 2011 - 2012 of Capital bikeshare system [203]. The raw dataset of the authors contains 8645 instances while the training set defined for the mini-challenge contains 6916 instances. Table 3.34 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 97.00%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.88%).

Table 3.34: Dataset 10: Rental Bikes Hourly. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	14	14	100	Canberra
Instances	8645	6916	88.889	Canberra
Data dimensionality	0.002	0.002	88.889	Canberra
Mean abs. Skewness	0.886	0.879	99.574	Canberra
Mean abs. Kurtosis	0.063	0.063	99.574	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean abs. linear correlation	0.282	0.259	97.724	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.0002	0.0001	99.997	Euclidean
Kurtosis Of Class	3.759	3.208	92.093	Canberra
Skewness Of Class	1.131	0.914	89.378	Canberra
		Similarity	97.008 %	

The original training set has a global similarity of 95.98 % respect to the training set cleaned by the CF as show Table 3.35. These datasets have a high global similarity due to CF applied two data cleaning task:

- Remove duplicate instances: this data cleaning task removed 0.1% of duplicate instances. Thus, the datasets have 99.99% of similarity between attributes.

- Dimensionality reduction: this data cleaning task discarded one attribute. Thus, the datasets have 96.29% of similarity between attributes.

- As a consequence of remove duplicate instances and dimensionality reduction, the meta-features mean absolute skewness, kurtosis, and linear correlation, instances and data dimensionality changed. mean absolute skewness and kurtosis contain the lowest similarities: 72.79% and 76.26% respectively.

Table 3.35: Dataset 10: Rental Bikes Hourly. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	14	13	96.296	Canberra
Instances	6916	6915	99.993	Canberra
Data dimensionality	0.002	0.002	96.304	Canberra
Mean abs. Skewness	0.879	0.503	72.796	Canberra
Mean abs. Kurtosis	0.063	0.039	76.261	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean abs. linear correlation	0.259	0.277	98.163	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.0001	0	99.986	Euclidean
Kurtosis Of Class	3.208	3.208	99.998	Canberra
Skewness Of Class	0.914	0.914	99.993	Canberra
		Similarity	95.986 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset “Rental Bikes Hourly” with the original training set and the training set cleaned by the CF. The authors of this dataset used Linear Regression (LR) and REP Tree. With aim to select the regression models statistically significant for each training set, we applied paired sample (t-test) [187] with $\rho = 0.5$. Table 3.36 and 3.37 present the MAE for the regression model built with the original training set and the training set cleaned by CF.

Table 3.36: Dataset 10: Rental Bikes Hourly (Training set). MAE measure for LR and REP Tree. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	MAE	LR	REP Tree
LR	0		(-)
REP Tree	4.58	(+)	

Table 3.37: Dataset 10: Rental Bikes Hourly (Training set cleaned by CF). AUC measure for LR and REP Tree. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the classifiers compared do not contain statistically significant differences.

	MAE	LR	REP Tree
LR	0		(-)
REP Tree	4.63	(+)	

For training sets, REP Tree classifiers are significantly better than Linear Regression classifier. The REP Tree built with original training set obtained 4.58 MAE, while the REP Tree generated with the training set cleaned by CF reached 4.63 MAE.

Thus, we validated the classifiers significantly better of the training sets (REP Tree classifiers) through test set defined for the mini-challenge. The test set contains 1383 instances. REP Tree of the original training set achieved the MAE lowest (3.51), however REP Tree of the training set cleaned by CF obtained a close MAE (3.70).

Dataset 5: Comments prediction in FB - 3 (Medium)

This dataset is oriented towards the comments prediction in a Facebook post [160]. The global similarity of this dataset corresponds to the average between the highest and lowest global similarity presented in Figure 3.25 for MAE measure. The raw dataset of the authors contains 121098 instances while the training set defined for the mini-challenge contains 96878 instances. Table 3.38 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 96.97%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.88%).

Table 3.38: Dataset 5: Comments prediction in FB - 3. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	53	53	100	Canberra
Instances	121098	96878	88.889	Canberra
Data dimensionality	0.0004	0.0005	88.889	Canberra
Mean abs. Skewness	15.981	15.203	97.505	Canberra
Mean abs. Kurtosis	0.302	0.287	97.505	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean abs. linear correlation	0.170	0.161	99.099	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.0005	0.0004	99.996	Euclidean
Kurtosis Of Class	369.528	447.031	90.509	Canberra
Skewness Of Class	14.811	16.056	95.967	Canberra
		Similarity	96.977 %	

The original training set has a global similarity of 95.15 % respect to the training set cleaned by the CF as show Table 3.39. The main differences between original training set and training set cleaned by CF are caused by application of the data cleaning tasks:

- Remove duplicate instances: 0.03 % of duplicate instances of the training set were removed. This data cleaning task reduces the similarity for meta-features: instances (99.97%) and Duplicate instances ratio (99.74%).
- Dimensionality reduction: this data cleaning task reduced the dimensionality of the dataset with the elimination of 16 attributes. Thus, the datasets have 82.22% of similarity between attributes and 82.24% of similarity for data dimensionality.

Table 3.39: Dataset 5: Comments prediction in FB - 3. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	53	37	82.222	Canberra
Instances	96878	96835	99.978	Canberra
Data dimensionality	0.001	0.000	82.244	Canberra
Mean abs. Skewness	15.203	17.238	93.727	Canberra
Mean abs. Kurtosis	0.287	0.466	76.214	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean abs. linear correlation	0.161	0.228	93.328	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.0003	0.000	99.744	Euclidean
Kurtosis Of Class	447.031	446.843	99.979	Canberra
Skewness Of Class	16.056	16.052	99.989	Canberra
		Similarity	95.155 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset "Comments prediction in FB - 3" with the original training set and the training set cleaned by the CF. The authors of this dataset used Multi Layer Perceptron (MLP), Radial Basis Function (RBF), REP Tree and Decision Tree M5. With aim to select the regression models statistically significant for each training set, we applied paired sample (t-test) [187] with $\rho = 0.5$. Table 3.40 and 3.41 present MAE for the regression models built with the original training set and the training set cleaned by CF.

Table 3.40: Dataset 5: Comments prediction in FB - 3 (Training set). Mean Absolute Error for MLP, RBF, REP Tree and M5. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the regression models compared do not contain statistically significant differences.

	MAE	MLP	RBF	REP Tree	M5
MLP	7.04		(+)	(-)	(-)
RBF	9.41	(-)		(-)	(-)
REP Tree	4.02	(+)	(+)		(.)
M5	3.89	(+)	(+)	(.)	

Table 3.41: Dataset 5: Comments prediction in FB - 3 (Training set cleaned by CF). Mean Absolute Error for MLP, RBF, REP Tree and M5. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the regression models compared do not contain statistically significant differences.

	MAE	MLP	RBF	REP Tree	M5
MLP	6.89		(+)	(-)	(-)
RBF	8.78	(-)		(-)	(-)
REP Tree	4.02	(+)	(+)		(.)
M5	3.90	(+)	(+)	(.)	

For training sets, M5 and REP Tree classifiers are significantly better than Multi Layer Perceptron and Radial Basis Function. However, M5 and REP Tree do not contain statistically significant differences. The M5 Tree built with original training set obtained 3.89 MAE, while REP Tree 4.02 MAE. In case of the training set cleaned by CF, M5 Tree obtained 3.90 MAE and REP Tree 4.02 MAE.

Thus, we validated the regression models significantly better of the training sets (M5 and REP Tree) through test set defined for the mini-challenge. The test set contains 24200 instances. M5 and REP trees of the training set cleaned by CF achieved the MAE lowest (5.46 and 5.52 respectively), compared with M5 (5.55 MAE) and REP Tree (5.59 MAE) of the original training set.

Dataset 15: Feedback Blogs Prediction

This dataset contains the lowest global similarity presented in Figure 3.25 for MAE measure. This dataset contain features extracted from blog posts for prediction of the number of comments in the upcoming 24 hours [208]. The raw dataset of the authors contains 52397 instances while the training set defined for the mini-challenge contains 41918 instances. Table 3.42 presents the local (for each meta-feature) and global similarity between the raw dataset of the authors and training set. The global similarity between the raw dataset of the authors and training set correspond to 95.87%, while the lowest local similarities are given by the meta-features: instances and data dimensionality (88.90%).

Table 3.42: Dataset 15: Feedback Blogs Prediction. Similarity between dataset of authors and training set

Meta-features	Authors	Training	Similarity (%)	Measure
Attributes	280	280	100	Canberra
Instances	52397	41918	88.9	Canberra
Data dimensionality	0.005	0.007	88.9	Canberra
Mean abs. Skewness	25.840	19.276	85.5	Canberra
Mean abs. Kurtosis	0.092	0.069	85.5	Arithmetic
Mean attribute entropy	0	0	1	Euclidean
Mean abs. linear correlation	0.070	0.074	99.6	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.061	0	97.2	Euclidean
Kurtosis Of Class	235.295	223.773	97.5	Canberra
Skewness Of Class	12.691	12.615	99.7	Canberra
		Similarity	95.877 %	

The original training set has a global similarity of 90.07 % respect to the training set cleaned by the CF as show Table 3.43. The low global similarity between these training sets is caused because the CF modified the original training set to apply the data cleaning tasks:

- Remove duplicate instances: 3.3 % of duplicate instances of the training set were removed (96.70% of local similarity in Duplicate instances ratio).
- Dimensionality reduction: this data cleaning task discarded 180 attributes

(60.00% of similarity between attributes). As a consequence of dimensionality reduction, the mean absolute skewness of the numeric attributes was decreased (53.94% of similarity between mean absolute skewness).

Table 3.43: Dataset 15: Feedback Blogs Prediction. Similarity between original training set and the training set cleaned by the CF

Meta-features	Training	Training CF	Similarity (%)	Measure
Attributes	280	120	60.00	Canberra
Instances	41918	40555	98.347	Canberra
Data dimensionality	0.007	0.003	61.397	Canberra
Mean abs. Skewness	19.276	7.120	53.945	Canberra
Mean abs. Kurtosis	0.069	0.059	92.577	Arithmetic
Mean attribute entropy	0	0	100	Euclidean
Mean abs. linear correlation	0.074	0.168	90.646	Euclidean
Equivalent num. of features	0	0	100	Canberra
Noise-signal ratio	0	0	100	Canberra
Missing values ratio	0	0	100	Euclidean
Duplicate instances ratio	0.033	0	96.70	Euclidean
Kurtosis Of Class	223.773	217.152	98.498	Canberra
Skewness Of Class	12.615	12.428	99.253	Canberra
		Similarity	90.071 %	

Subsequently, we trained the same algorithms proposed by authors of the dataset "Feedback Blogs Prediction" with the original training set and the training set cleaned by the CF. The authors of this dataset used Decision Tree M5, REP Tree, Linear Regression (LR) and Multi Layer Perceptron (MLP). With aim to select the regression models statistically significant for each training set, we applied paired sample (t-test) [187] with $\rho = 0.5$. Table 3.44 and 3.45 present MAE for the regression models built with the original training set and the training set cleaned by CF.

Table 3.44: Dataset 15: Feedback Blogs Prediction (Training set). Mean Absolute Error for M5, REP Tree, LR and MLP. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the regression models compared do not contain statistically significant differences.

	MAE	M5	REP Tree	LR	MLP
M5	5.61		(.)	(+)	(+)
REP Tree	5.64	(.)		(+)	(+)
LR	9.39	(-)	(-)		(-)
MLP	10.60	(-)	(-)	(+)	

Table 3.45: Dataset 15: Feedback Blogs Prediction (Training set cleaned by CF). MAE for M5, REP Tree, LR and MLP. The signals below the diagonal represent the results to apply the t-test. The symbol (+)/(-) indicates that the classifier of the row i is significantly better/worst than classifier of the column j. The symbol (.) means that the regression models compared do not contain statistically significant differences.

	MAE	M5	REP Tree	LR	MLP
M5	5.51		(-)	(+)	(.)
REP Tree	5.66	(+)		(+)	(+)
LR	9.47	(-)	(-)		(-)
MLP	8.37	(.)	(-)	(+)	

For original training set, M5 and REP Tree are significantly better than Multi Layer Perceptron and Linear Regression. However, M5 and REP Tree do not contain statistically significant differences. In case of the training set cleaned by CF, REP Tree is the regression model significantly better. The M5 built with original training set obtained 5.61 MAE, while REP Tree 5.64 MAE. For training set cleaned by CF, REP Tree obtained 5.66 MAE.

Thus, we validated the regression models significantly better (M5 and REP Tree for original training set, and REP Tree for training set cleaned by CF) through test set defined for the mini-challenge. The test set contains 10479 instances. REP tree of the training set cleaned by CF achieved the MAE lowest (8.12), follow by

REP tree and M5 of the original training set with 8.41 and 8.97 MAE respectively.

In summary, 2/3 regression mini-challenges, the models generated by the datasets cleaned by CF reached the lowest *MAE*.

3.6 Summary

This chapter presents the conceptual data quality framework for classification and regression tasks. We adapted the methodology of [21] for building our conceptual framework (CF) following the phases:

1. **Mapping the selected data sources:** we identified the data quality issues presented in classification and regression tasks. We reviewed four relevant methodologies: *Knowledge Discovery in Databases (KDD)* [116], *Cross Industry Standard Process for Data Mining (CRISP-DM)* [13], *Sample, Explore, Modify, Model and Assess (SEMMA)* [23] and *The Data Science Process* [117]. Also, we found a taxonomy of data quality challenges in empirical software engineering (ESE), based on an literature review [120]. *Noise*, *missing values*, *outliers*, *high dimensionality*, *inconsistency*, *redundancy*, *amount of data*, *heterogeneity*, and *timeliness* were the data quality issues found in the knowledge discovery methodologies and ESE taxonomy.

2. **Understanding the selected data:** in this phase we explained the data quality found in the knowledge discovery methodologies and ESE taxonomy.

3. **Identifying and categorizing components:** we organized and filtered the data quality issues according to their meaning:

    - *Inconsistency*, *redundancy* and *timeliness* were renamed as *mislabelled class*, *duplicate instances* and *data obsolescence*.
    - We considered kinds of *noise*: *missing values*, *outliers*, *high dimensionality*, *imbalanced class*, *mislabelled class* and *duplicate instances*.
    - *Amount of data*, *heterogeneity* and *data obsolescence* are issues of recollection data process. These data quality issues were classified in a new category called *Provenance*.

4. **Integrating components:** we defined the data cleaning tasks to address the data quality issues. Subsequently, we proposed the conceptual framework (CF) based on the integration of the data cleaning tasks.

5. **Validation:** the conceptual framework (CF) was evaluated through 48 datasets (28 datasets for classification and 20 for regression) of the UCI Repository of Machine Learning Databases [158]. The cleaned datasets by our conceptual framework were used to train the same algorithms proposed by authors of UCI datasets. For classification datasets, 85.71% of the models (generated by the datasets cleaned by CF) achieve the highest *Precision* and *AUC* than models proposed by datasets authors. In case of regression datasets, 90% of the models reach MAE less than models proposed by datasets authors. With respect to mini-challenges, 4/6 classification mini-challenges, the classifiers generated by the datasets cleaned by CF achieved the highest *Accuracy* and *AUC*, while 2/3 regression mini-challenges, the models generated by the datasets cleaned by CF reached the lowest *Mean Absolute Error*.

In summary, the effort in data preparation of the dataset of authors can be addressed by the conceptual framework. Our approach offers a general data cleaning solution tested on 56 datasets of the UCI Repository.

4. Data Cleaning Ontology

This chapter explains the proposed ontology called Data Cleaning Ontology (DCO). This ontology gathers the knowledge of the data cleaning algorithms to solve the data quality issues, besides of set of rules that allow to know the data cleaning methods respect to data cleaning approach. DCO supports the Case–based reasoning (Chapter 5) in the case representation, and reuse phase.

Initially, we searched a methodology for building DCO. Thus, we reviewed the work of [211], which compares six methodologies to build ontologies: Uschold and Kings [212], METHONTOLOGY [82], On-To-Knowledge [213], Noy and McGuinness [214], TERMINAE [215] and Termontography [216]. The ontology methodologies were compared based on next criteria:

- C1: Intended audience. Persons that use the ontology methodology.
- C2: Level of detail (1-5). The ontology methodology recommends the methods and techniques to use in order to perform the different activities.
- C3: Associated software application. The methodology recommends to use a software application to build the ontology.
- C4: Conceptualization phase. The methodology organizes and structures the knowledge, independent from the knowledge representation paradigms and ontology languages. The representations must be comprehensible by domain experts and ontology developers through diagrams and tables.

Table 4.1 shows the comparison of methodologies based on four criteria.

Table 4.1: Comparison of methodologies to build ontologies. Source: [211]

Methodology	**C1**	**C2**	**C3**	**C4**
Uschold and Kings [212]	Ontology developers	3	No	No
METHONTOLOGY [82]	Ontology engineers and researchers	5	WebODE and Protégé	Yes
On-To-Knowledge [213]	Ontology developers	4	OntoStudio	Yes
Noy and McGuinness [214]	Ontology developers	5	Protégé	No
TERMINAE [215]	Knowledge engineers and terminologists	4	Terminae	Yes
Termontography [216]	Ontology builders, terminographers and lexicographers	3	Termontography Workbench	No

Based on the results of Table 4.1 , METHONTOLOGY accomplishes the four criteria. This methodology is the most suitable to build ontologies due high level of detail of instructions, good representation through diagrams, tables and compatibility with popular ontology editors. Thus, we selected METHONTOLOGY [82] as the methodology to create DCO.

METHONTOLOGY defines five phases: glossary of terms, concept taxonomies, ad hoc binary relation diagrams, concept dictionary, and rules. Next, we describe the way DCO was created following the phases mentioned above.

4.1 Build glossary of terms

In this task are identified the set of terms to be included on the *Data cleaning ontology* (their natural language definition, and their synonyms and acronyms). First, we identified the meaning and type of term, as show the Table 4.2.

Table 4.2: Description and type (Class, Instance, Attribute) of the terms of *Data cleaning ontology*

Name	**Description**	**Type**
Dataset	Collection of data organized by rows and columns	Class
Attribute	Feature of a dataset	Class
Target	Class of a dataset	Class
Data quality issue	Problem presented in a dataset	Class
Missing values	Refers to when one variable or attribute does not contain any value	Instance
Outliers	These are observations which deviate much from other observations	Instance
Mislabelled class	Contradictory instances	Instance
Imbalanced class	When a dataset exhibits an unequal distribution between its classes	Instance
Duplicate instances	Represent instances with same values	Instance
High dimensionality	When dataset contains a large number of features.	Instance
Data cleaning task	Task to address a data quality issue	Class
Imputation	Data cleaning task to fill missing values	Class
Outliers detection	Data cleaning task to detect outliers	Class
Label correction	Data cleaning task to detect instances with the mislabelled class	Class
Classes balancing	Data cleaning task to balance the instances of the minority class	Class
Remove duplicate instances	Data cleaning task to remove duplicate instances	Class
Dimensionality reduction	Data cleaning task to reduce the dataset dimensionality finding a subset of useful features to represent the dataset	Class
Model	Representation of a dataset from a mathematical function	Class
Performance	Refers to performance measures of the models	Class

Subsequently, we verified the synonyms and acronyms of the terms as show the Table 4.3.

Table 4.3: Synonyms and acronyms of the terms of *Data cleaning ontology*

Name	**Synonyms**	**Acronyms**
Dataset	–	–
Attribute	Feature / Variable	Att / Var
Target	Class / Dependent variable	Att / Var
Data quality issue	Data quality problem	DQ issue
Missing values		–
Outliers	–	–
Mislabelled class	–	–
Imbalanced class	–	–
Duplicate instances	–	–
High dimensionality	–	
Data cleaning task	–	DC Task
Imputation	Synthetic data	–
Outliers detection	–	–
Label correction	–	–
Classes balancing	–	–
Remove duplicate instances	–	–
Dimensionality reduction	Feature selection	FS
Model	Classifier	–
Performance	–	–

We defined 23 sub classes of the classes presented above. It is shown through taxonomies.

4.2 Build concept taxonomies

In this task, concepts taxonomies are created from the glossary of terms. We defined two general taxonomies from classes: *Attribute* and *Data cleaning task*.

An *Attribute* can be *Numeric* with continuous values or *Nominal* with discrete values as shown Figure 4.1

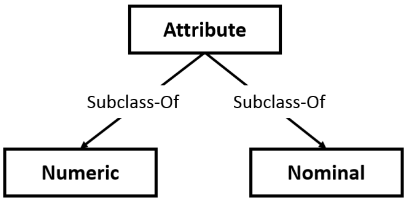


Figure 4.1: Taxonomy of the Concept: Attribute. The white square depicts the classes. The solid line represents a hierarchical relation.

In Figure 4.2 are presented the type of *Data cleaning tasks*: *Imputation*, *Outliers Detection*, *Classes balancing*, *Label correction*, *Dimensionality Reduction*, and *Remove duplicate instances*.

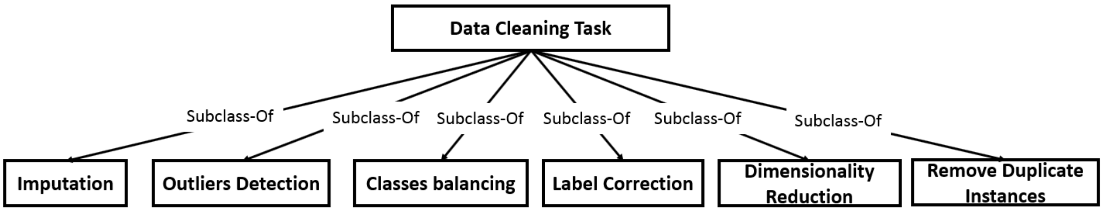


Figure 4.2: Taxonomy of the Concept: Data cleaning task. The white square depicts the classes. The solid line represents a hierarchical relation.

Sub-classes of *Data cleaning algorithm* have itself approaches which are presented below:

- *Imputation* is resolved through approaches: *Imputation Based On Non Missing Attributes*, *Deletion*, *Hot Deck Imputation*, *Imputation Based On Missing Attributes*. Figure 4.3 are presented the *Imputation* approaches:

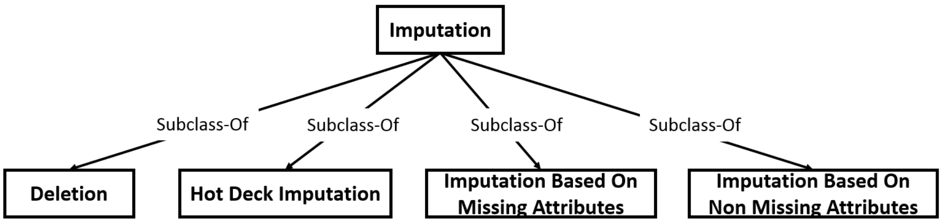


Figure 4.3: Taxonomy of the Concept: Imputation. The white square depicts the classes. The solid line represents a hierarchical relation.

- For *Outliers Detection* are used approaches based on *Clustering* or *High Dimensional*. Figure 4.4 are depicted:

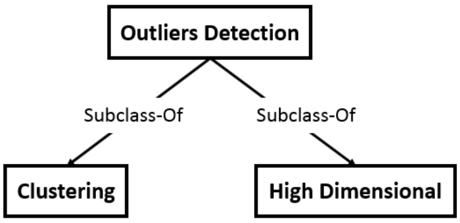


Figure 4.4: Taxonomy of the Concept: Outlier Detection. The white square depicts the classes. The solid line represents a hierarchical relation.

- Figure 4.5 shows the approaches to *Classes balancing*: *Over Sampling* or *Under Sampling*.

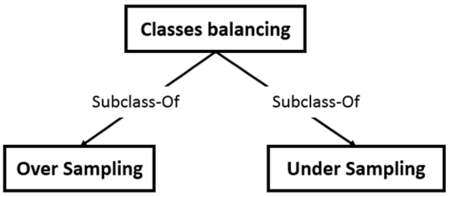


Figure 4.5: Taxonomy of the Concept: Classes balancing. The white square depicts the classes. The solid line represents a hierarchical relation.

- *Label correction* is addressed in two ways: approaches based on *Threshold* or *Classification* algorithms. They are shown in Figure 4.6:

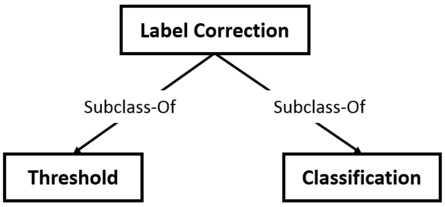


Figure 4.6: Taxonomy of the Concept: Label correction. The white square depicts the classes. The solid line represents a hierarchical relation.

- Approaches as *Embedded*, *Filter*, *Projection* and *Wrapper* are used to *Dimensionality Reduction*. Figure 4.7 list the approaches:

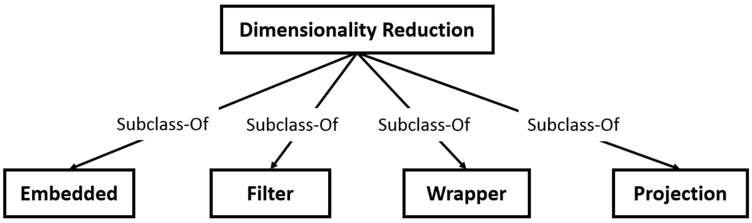


Figure 4.7: Taxonomy of the Concept: Dimensionality Reduction. The white square depicts the classes. The solid line represents a hierarchical relation.

- *Remove duplicate instances* does not contain sub-classes.

4.3 Build ad hoc binary relation diagrams

In this task, we establish ad hoc relationships between classes. Figure 4.8 presents seven binary relations among six classes.

- A *Dataset* (1..1) has *Data Quality Issue* (1..*): datasetHasDQIssue
- A *Data Quality Issue* (1..*) is resolved with *Data cleaning task* (1..*): DQIssueIsresolvedWithDCTask
- A *Dataset* (1..1) uses *Data cleaning tasks* (1..*): datasetUsesDCTask
- An *Attribute* (1..*) is part of a *Dataset* (1..1): attributeIsPartOfDataset
- An *Attribute* (1..*) has *Data Quality Issue* (1..*): attributeHasDQIssue
- A *Model* (1..*) is built with *Dataset* (1..1): modelIsBuiltWithDataset
- A *Model* (1..1) has *Performance* (1..*): modelHasPerformance

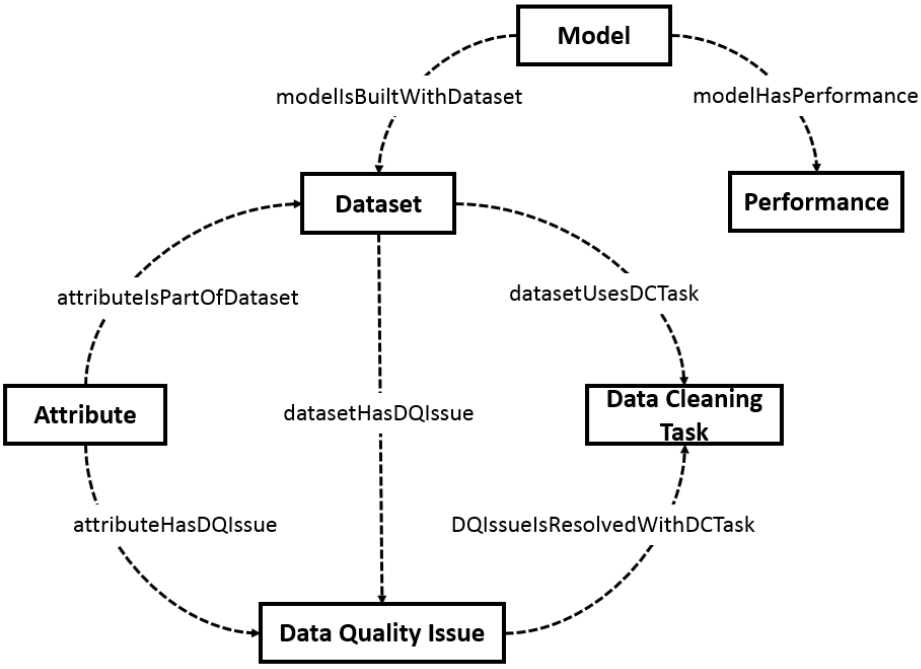


Figure 4.8: Binary relations of Data cleaning ontology. The white square depicts the classes. The dotted line represents a relation between two classes.

In addition, Table 4.4 presents the inverse relation of the ad hoc binary relations.

Table 4.4: Inverse relations of ad hoc binary relations of the *Data cleaning ontology*

Relation name	Inverse relation
Dataset has *Data Quality Issue*	*Data Quality Issue* is presented in a *Dataset*
Data Quality Issue is resolved with *Data cleaning task*	*Data cleaning task* resolved *Data Quality Issue*
Dataset uses *Data cleaning tasks*	*Data cleaning tasks* is applied in *Dataset*
An *Attribute* is part of a *Dataset*	*Dataset* contains *Attributes*
An *Attribute* has *Data Quality Issue*	*Data Quality Issue* is presented in a *Attribute*
A *Model* is built with *Dataset*	A *Dataset* is used to build a *Model*
A *Model* has *Performance*	–

4.4 Build concept dictionary

This task explains the instances and features of the classes. We present three subsections, the first describes the classes: *Dataset* and *Data quality*, follow by *Data cleaning task* class and finally, the classes: *Model* and *Performance*.

4.4.1 Dataset and Data Quality Issue

Dataset class presents twelve features related with instances, attributes, data dimensionality, missing values ratio, Duplicate instances ratio, mean absolute linear correlation, mean attribute entropy, mean absolute skewness, equivalent number of features, noise–signal ratio, mean absolute kurtosis and mean mutual information. *Dataset* class contains 56 instances. These instances correspond to UCI datasets [158] selected to evaluate the conceptual framework.
For the class, *Attribute* of dataset was defined the features: missing values ratio and correlation coefficient. When the attribute is *Numeric*, three features were selected: Candidate outliers, Kurtosis, and Skewness, in case of the attribute is Nominal, three features were considered: Normalized Entropy, Mutual information, Labels. For *Target* variable were used the same features of an *Attribute* (*Numeric* or *Nominal*). When the *Target* variable is *Nominal*, the Imbalance ratio is considered. To select the meta-features, we reviewed several works which were analyzed in Subsection 5.1.1.

The *Data Quality Issue* class is composed of the instances: *missing values*, *outliers*, *imbalanced class*, *mislabeled class*, *duplicate instances* and *high dimensional spaces*. Table 4.5 presents a summary of class features for *Dataset*, *Attribute*, *Nominal*, *Numeric* and *Data Quality Issue*.

Table 4.5: Concept dictionary of: *Dataset*, *Attribute*, *Nominal*, *Numeric* and *Data Quality Issue*

Class name	**Class features**
Dataset	Instances, attributes, data dimensionality, missing values ratio, duplicate instances ratio, mean absolute linear correlation, mean attribute entropy, mean absolute skewness, equivalent number of features, noise–signal ratio, mean absolute kurtosis and mean mutual information
Attribute	Missing values ratio, correlation coefficient
Nominal	Normalized entropy, mutual information, labels
Numeric	Candidate outliers, kurtosis, skewness
Data Quality Issue	Missing values, outliers, imbalanced class, mislabeled class, duplicate instances, high dimensionality

4.4.2 Data cleaning task

In this subsection, we show the instances of the data cleaning tasks. For example, Table 4.6 shows the instances of *Imputation*. Thus *Deletion* is represented by instances (algorithms): list wise deletion, pair wise deletion, while *Hot Deck Imputation*: last observation carried forward. In case of *Imputation Based On Missing Attributes* by the instances (algorithms): mean, median, mode and *Imputation Based On Non Missing Attributes* by models: linear, logistic, random forest and bayesian.

Table 4.6: Concept dictionary of: *Imputation*, *Deletion*, *HotDeckImputation*, *ImputationBasedOnMissingAttributes* and *ImputationBasedOnNonMissingAttributes*

Class name	**Instances**
Imputation	–
Deletion	list wise deletion, pair wise deletion
Hot Deck Imputation	last observation carried forward
Imputation Based On Missing Attributes	mean, median, mode
Imputation Based On Non Missing Attributes	bayesian linear regression, linear regression, logistic regression, random forest

Table 4.7 presents the attributes of the classes *ImputationBasedOnMissingAttributes* and *HotDeckImputation*.

Table 4.7: Attributes of the Classes: ImputationBasedOnMissingAttributes and HotDeckImputation

Attribute name	Description	Class	Value type
Iteration	Number of iterations	Imputation Based On Missing Attributes	Integer
Validation	Method of validation (CV, LOOCV, etc.)	Imputation Based On Missing Attributes	Integer
Number	Number of folds	Imputation Based On Missing Attributes	Integer
Method	Last observation, first observation	Hot Deck Imputation	String

Table 4.8 gathers algorithms for *Outliers Detection*. Density-based spatial clustering of applications with noise (dbscan, local outlier factor and ordering points to identify the clustering structure (optics) are algorithms based on *Clustering*. In *High Dimensional* spaces are used algorithms as: angle based outlier degree, grid based subspace outlier, and sub space outlier degree.

Table 4.8: Concept dictionary of: *Outliers Detection*, *Density*, *High Dimensional* and *Removing of duplicate instances*

Class name	Instances
Outliers Detection	–
Clustering	dbscan, local outlier factor, optics
High Dimensional	angle based outlier degree, grid based subspace outlier, sub space outlier degree
Remove duplicate instances	standard duplicate elimination

Table 4.9 presents the attributes of the classes *Clustering* and *High dimensional*.

Table 4.9: Attributes of the Classes: Clustering and High dimensional

Attribute name	Description	Class	Value type
Eps	Epsilon	Clustering	Float
MinPts	Minimum number of points to consider a dense region	Clustering	Integer
Distance	Euclidean, Mahalanobis, Canberra, etc.	Clustering, High dimensional	String
K	Number of neighbors	Clustering, High dimensional	Integer

Table 4.10 encompasses instances of the approaches of *Classes balancing* and *Label correction*. Random over sampling and smote are algorithms of *Over sampling*, while condensed nearest neighbor rule, edited nearest neighbor rule, neighborhood cleaning rule, one side selection, random under sampling, tomek link of *Under Sampling* approach. In *Label correction* are commonly used *Classification* algorithms as c4.5, k nearest neighbor, support vector machine and *Threshold* as entropy conditional distribution, least complex correct hypothesis.

Table 4.10: Concept dictionary of: *ClassesBalancing*, *OverSampling*, *UnderSampling*, *LabelCorrection*, *Classification* and *Threshold*

Class name	Instances
Classes balancing	–
Over sampling	random over sampling, smote
Under sampling	condensed nearest neighbor rule, edited nearest neighbor rule, neighborhood cleaning rule, one side selection, random under sampling, tomek link
LabelCorrection	–
Classification	decision tree, k nearest neighbor, support vector machine
Threshold	entropy conditional distribution, least complex correct hypothesis

Table 4.11 presents the attributes of the classes *OverSampling*, *UnderSampling* and *Threshold*.

Table 4.11: Attributes of the Classes: OverSampling, UnderSampling and Threshold

Attribute name	Description	Class	Value type
K	Number of neighbors	OverSampling, UnderSampling	Integer
PercOver	Percentage of instances to create	OverSampling	Float
Method	Dirichlet probability distribution, empirical probability distribution	Threshold	String

Table 4.12 contains *Filter*, *Projection* and *Wrapper* algorithms for *Dimensionality Reduction*. Measures as chi–squared, gain ratio, information gain, Pearson correlation, and spearman correlation belong to *Filter* approach. Principal component analysis is an algorithm based on *Projection*, while sequential backward elimination and sequential forward selection are algorithms based on *Wrapper* approach.

Table 4.12: Concept dictionary of: *Dimensionality Reduction*, *Embedded*, *Filter*, *Projection* and *Wrapper*

Class name	Instances
Dimensionality Reduction	–
Embedded	–
Filter	chi–squared, gain ratio, information gain, Pearson correlation, spearman correlation
Projection	principal component analysis
Wrapper	sequential backward elimination, sequential forward selection

Table 4.13 presents the attributes of the classes *Wrapper* and *Embedded*.

Table 4.13: Class attributes of the *Data cleaning ontology*

Attribute name	Description	Class	Value type
Classifier	Decision tree, neural network, support vector machine, etc.	Wrapper, Embedded	String
Validation	Method of validation (CV, LOOCV, etc.)	Wrapper	String
Number	Number of folds	Wrapper	Integer

4.4.3 Model and Performance

In Table 4.14, we present the features of the classes: *Model* and *Performance*.

Table 4.14: Concept dictionary of: *Model* and *Performance*

Class name	Class features
Model	Name, knowledge discovery task,
Performance	Measure, Value, experiment description

The *Model* class is described by the name of model (e.g. decision tree, neural network, etc.), and the knowledge discovery task (classification or regression), while the *Performance* class presents the assessment measure (e.g. precision, recall, mean absolute error, etc.) and the experiment description (e.g. cross validation, test set, etc.).

4.5 Describe rules

We used Semantic Web Rule Language (SWRL) to create the rules of *Data cleaning ontology*. SWRL is a proposal to combine OWL and RuleML. The rules are expressed regarding of OWL concepts (classes, attributes, instances) and saved as part of the ontology. These include a high-level abstract syntax for Horn-like rules [217]. The rules syntax have the form: $antecedent \rightarrow consequent$, where the antecedent and consequent are conjunctions of atoms $a_1 \wedge ... \wedge a_n$ and functions $f_1(?\,a_1, ?\,a_2) \wedge ... \wedge f_n(?\,a_n)$. The variables are represented through question mark (e.g., $?\,a_1$).

We built 19 rules to detect data quality issues and select the available algorithms of data cleaning approaches. Below are presented the DCO rules.

4.5.1 Data quality issues

First, we define the rules of data quality issues. For example, *Dataset* with missing values ratio (mv_att) greater than 0% has *missing values* (Rule 4.1):

$$\begin{gathered} Dataset(?\,ds) \wedge mv\_att(?\,ds, ?\,mv) \wedge swrlb\colon greaterThan(?\,mv, 0) \\ \rightarrow datasetHasDQIssue(?\,a, missingValues) \end{gathered} \tag{4.1}$$

An *Attribute* of *Dataset* with candidate *outliers* is represented when the outliers ratio (out_att) is greater than 0% (Rule 4.2):

$$\begin{gathered} Dataset(?\,ds) \wedge Attribute(?\,a) \wedge attributeIsPartOfDataset(?\,a, ?\,ds) \\ \wedge\, out\_att(?\,a, ?\,out) \wedge swrlb\colon greaterThan(?\,out, 0) \\ \rightarrow datasetHasDQIssue(?\,ds, outliers) \end{gathered} \tag{4.2}$$

Dataset with *imbalanced class* occurs when imbalance ratio is greater than 1 (Rule 4.3):

$$\begin{gathered} Dataset(?\,ds) \wedge imbalanceRatio(?\,ds, ?\,ir) \wedge swrlb\colon greaterThan(?\,ir, 1) \\ \rightarrow datasetHasDQIssue(?\,a, imbalancedClass) \end{gathered} \tag{4.3}$$

Similarly, a *Dataset* with duplicate instances ratio (dupIns_att) greater than 0 contains *duplicate instances* (Rule 4.4):

$$\begin{gathered} Dataset(?\,ds) \wedge dupIns\_att(?\,ds, ?\,di) \wedge swrlb\colon greaterThan(?\,di, 0) \\ \rightarrow datasetHasDQIssue(?\,a, duplicateInstances) \end{gathered} \tag{4.4}$$

In case of *mislabeled classes* and *high dimensionality* we can not know to priori whether a *Dataset* contains these data quality issues.

4.5.2 Data cleaning tasks

Once defined the rules of data quality issues, we built the rules to select the available algorithms of data cleaning approaches.

4.5.2.1 Imputation

In case of *Imputation*, we built 4 rules. The Rule 4.5 presents the *Deletion* algorithms.

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, missingValues) \\ &\quad \wedge Deletion(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b) \end{aligned} \tag{4.5}$$

The Rule 4.6 lists the data cleaning algorithms of *Imputation Based On Non Missing Attributes* and Rule 4.7 shows the data cleaning algorithms of *Imputation Based On Missing Attributes*.

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, missingValues) \\ &\quad \wedge ImputationBasedOnNonMissingAttributes(?\,b) \\ &\qquad \rightarrow datasetUsesDCTask(?\,ds, ?\,b) \end{aligned} \tag{4.6}$$

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, missingValues) \\ &\quad \wedge ImputationBasedOnMissingAttributes(?\,b) \\ &\qquad \rightarrow datasetUsesDCTask(?\,ds, ?\,b) \end{aligned} \tag{4.7}$$

The methods of *Hot Deck Imputation* are listed through the Rule 4.8.

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, missingValues) \\ &\quad \wedge HotDeckImputation(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b) \end{aligned} \tag{4.8}$$

4.5.2.2 Outliers detection

The *Clustering* methods for Outlier detection is given by Rule 4.9. The same structure was used for *HighDimensional* approach (Rule 4.10).

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, outliers) \\ &\quad \wedge Clustering(?\,b) \wedge datasetUsesDCTask(?\,ds, ?\,b) \end{aligned} \tag{4.9}$$

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, outliers) \\ &\quad \wedge HighDimensional(?\,b) \wedge datasetUsesDCTask(?\,ds, ?\,b) \end{aligned} \tag{4.10}$$

4.5.2.3 Classes balancing

The rule 4.11 shows the *Oversampling* methods. For *Undersampling* approach, we used the same structure (Rule 4.12).

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, imbalancedClass) \\ &\quad \wedge OverSampling(?\,over) \rightarrow datasetUsesDCTask(?\,ds, ?\,over) \end{aligned} \tag{4.11}$$

$$\begin{aligned} &Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, imbalancedClass) \\ &\quad \wedge Undersampling(?\,over) \rightarrow datasetUsesDCTask(?\,ds, ?\,over) \end{aligned} \tag{4.12}$$

4.5.2.4 Dimensionality reduction

To invoke a dimensionality reduction approach, we defined the structure of the Rule 4.13. In this case, the Rule 4.13 lists the Wrapper methods:

$$\begin{gathered}Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, highDimensionality)\\ \wedge\, Wrapper(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b)\end{gathered} \tag{4.13}$$

, similarly the Embedded rule (Rule 4.14):

$$\begin{gathered}Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, highDimensionality)\\ \wedge\, Embedded(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b)\end{gathered} \tag{4.14}$$

in case of the Filter rule (Rule 4.15):

$$\begin{gathered}Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, highDimensionality)\\ \wedge\, Filter(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b)\end{gathered} \tag{4.15}$$

and the Projection rule (Rule 4.16):

$$\begin{gathered}Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, highDimensionality)\\ \wedge\, Projection(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b)\end{gathered} \tag{4.16}$$

4.5.2.5 Label correction

In addition, we defined rules for label correction, as shown the Rule 4.17 for *Threshold* and Rule 4.18 for *Classification*:

$$\begin{gathered}Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, mislabeledClass)\\ \wedge\, Threshold(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b)\end{gathered} \tag{4.17}$$

$$\begin{gathered}Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, mislabeledClass)\\ \wedge\, Classification(?\,b) \rightarrow datasetUsesDCTask(?\,ds, ?\,b)\end{gathered} \tag{4.18}$$

4.5.2.6 Remove duplicate instances

As *Remove duplicate instances* does not contain sub-classes, we built the Rule 4.19 to invoke the *Standard Duplicate Elimination* algorithm.

$$\begin{gathered}Dataset(?\,ds) \wedge datasetHasDQIssue(?\,ds, duplicateInstances)\\ \wedge\, RemoveDuplicateInstances(?\,b)\\ \rightarrow datasetUsesDCTask(?\,ds, ?\,b)\end{gathered} \tag{4.19}$$

4.6 Ontology Editor

The *Data cleaning ontology* was modeled in the Ontology editor *Protégé* as show Figure 4.9.

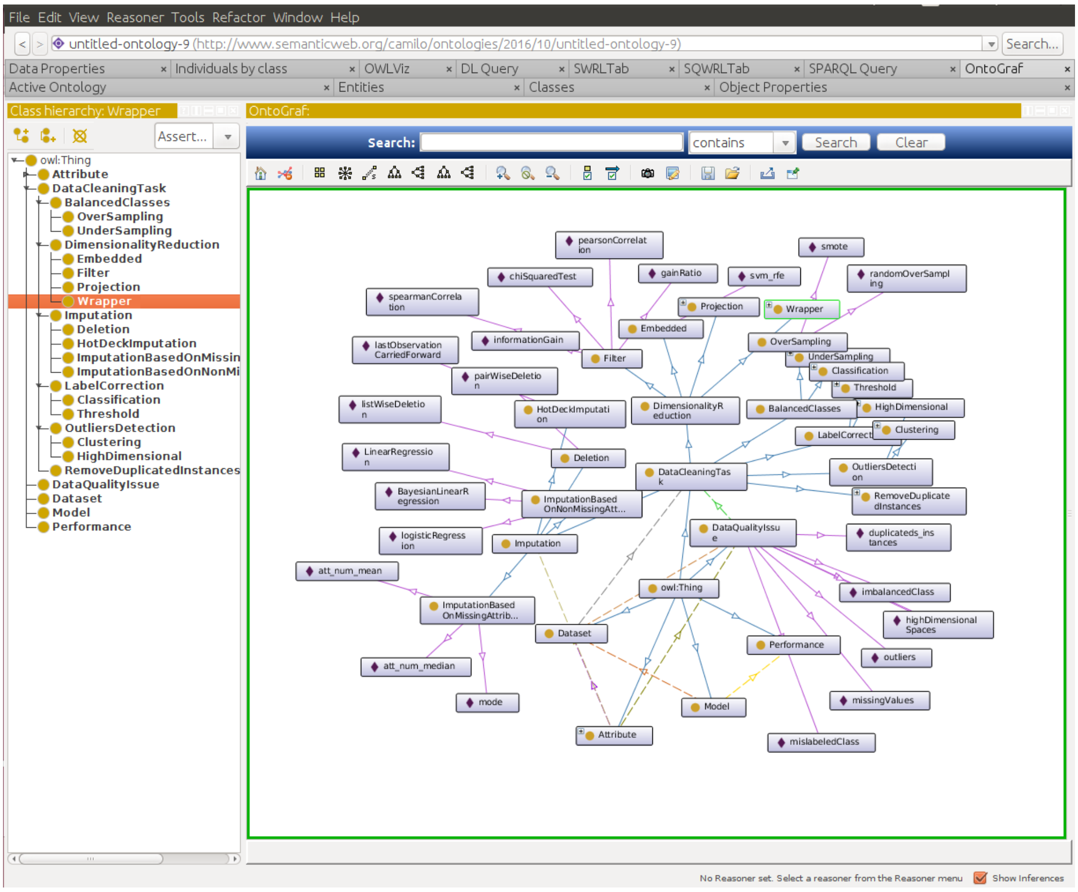


Figure 4.9: Screenshot of DCO in ontology editor: Protégé

Protégé is software tool based on Java with license open source. The *Protégé* software tool supports two types of modeling ontologies: (i) Protégé-Frames and (ii) Protégé-OWL editors. *Protégé* supports different formats including RDF, OWL, and XML Schema [115]. In Figure 4.9 is presented the classes of *Data cleaning ontology* through Hierarchy Tool Window, also a graphical representation of the classes and individuals in the Graphical User Interface (GUI): OntoGraf. In addition, *Protégé* offers several GUI to show the properties of *Data cleaning ontology*, for example the relations of the ontology (GUI: Object properties), the instances (GUI: Individuals) and features (GUI: Data properties) of the ontology , SWRL rules (GUI: SWRL tab), etc. The *Data cleaning ontology* is available in the URL: `http://artemisa.unicauca.edu.co/~dcorrales/ontology/DCO_v1.3.owl`.

4.7 Summary

In this chapter we described the *Data cleaning ontology* (DCO) to represent the knowledge of data quality issues in classification and regression tasks and data cleaning tasks to address the data quality issues. First, we reviewed the work of [211], which compares six methodologies to build ontologies: Uschold and Kings [212], METHONTOLOGY [82], On-To-Knowledge [213], Noy and McGuinness [214], TERMINAE [215] and Termontography [216]. Based on analysis of the authors [211], we selected METHONTOLOGY [82] as the methodology to create DCO. METHONTOLOGY defines five phases:

1. **Build glossary of terms:** in this phase were identified the set of terms included on the *Data cleaning ontology* as *Dataset, Attribute, Data quality issue, Data cleaning task, Classes balancing, Dimensionality reduction, Imputation, Label correction, Outliers detection, Remove duplicate instances, Outliers detection, Model* and *Performance*.

2. **Build concept taxonomies:** we presented seven taxonomies for the classes *Attribute, Data cleaning task, Imputation, Outliers Detection, Classes balancing, Label correction*, and *Dimensionality Reduction*.

3. **Build ad hoc binary relation diagrams:** in this phase were defined the relations between DCO classes:

   - A *Dataset* has *Data Quality Issue*.
   - A *Data Quality Issue* is resolved with *Data cleaning task*.
   - A *Dataset* uses *Data cleaning tasks*.
   - An *Attribute* is part of a *Dataset*.
   - An *Attribute* has *Data Quality Issue*.
   - A *Model* is built with a *Dataset*.
   - A *Model* has *Performance*.

4. **Build concept dictionary:** this phase described the instances and features of the DCO classes. We presented three subsections, the first described the classes: *Dataset* and *Data quality*, followed by *Data cleaning task* class and finally, the classes: *Model* and *Performance*.

5. **Describe rules:** the rules were built in Semantic Web Rule Language (SWRL). We built 19 rules to detect data quality issues (4 rules) and select the available algorithms of data cleaning approaches (15 rules).

Additionally, we shown the main classes of the *Data cleaning ontology* in ontology editor: Protégé. We highlighted the main the Graphical User Interface as OntoGraf, Hierarchy Tool Window, Object properties, Individuals, Data properties and SWRL tab.

5. Case-based reasoning for data cleaning

The construction of a Case-based reasoning (CBR) for data cleaning focused to non-specialist users is a challenging and complex endeavor. Consequently, the construction of a CBR involves a host of important design considerations (i.e. case representation, filter mechanisms, similarity measures, reuse and revision of cases, etc) [104]. This chapter presents the Case-based reasoning system for data cleaning. The aim of our CBR is to recommend data cleaning algorithms to the inexpert data analyst with the goal of preparing the dataset for classification and regression tasks.

First, we explain the case-base construction and case representation. A case is represented by problem and solution . The problem space is defined from a set of dataset meta-features, while the solution space by a set of data cleaning algorithms used to address the data quality issues found in the dataset. In addition, the CBR is composed by three stages:

- Retrieval phase: where the most similar case to a new case is retrieved.
- Reuse phase: similar solutions to the solution of the retrieved case are proposed by data cleaning ontology (Chapter 4).
- Retain phase: the new case is assessed for retention, considering three data quality dimensions: Accuracy, Completeness, and Validity.

Figure5.1 presents the CBR proposed.

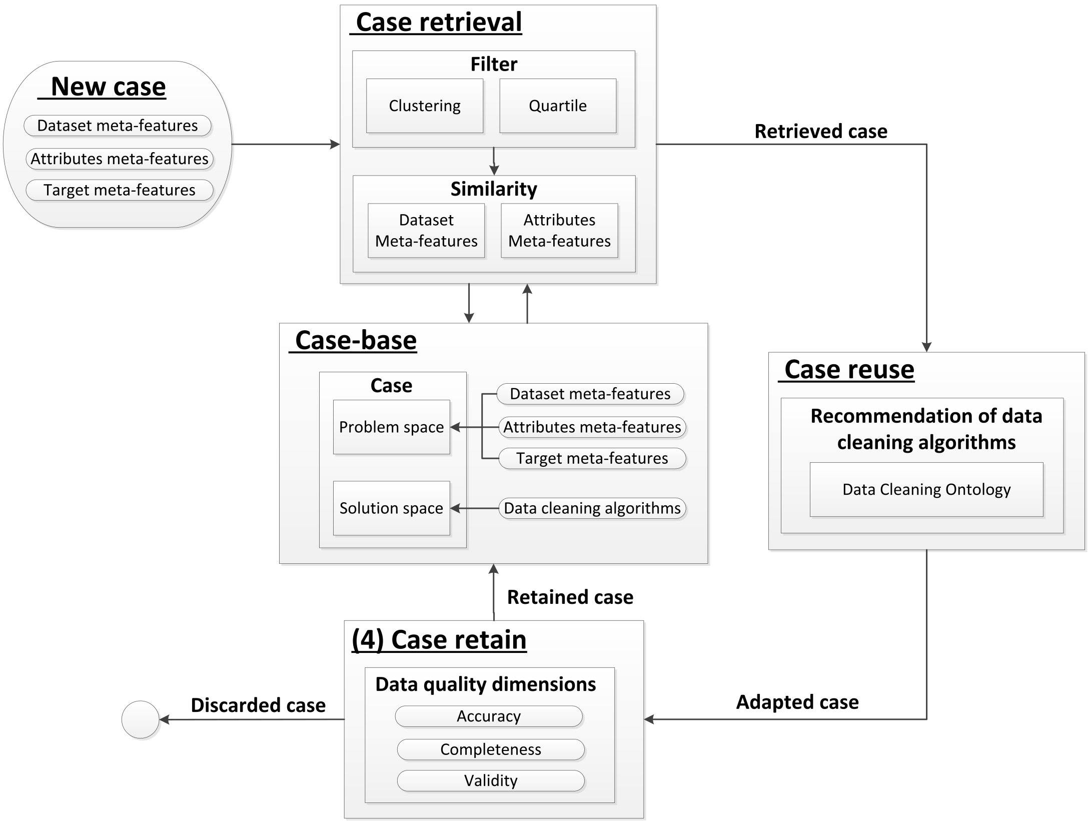


Figure 5.1: CBR for data cleaning in knowledge discovery tasks. The CBR is composed by case-base, and three phases: case retrieval, case reuse and case retain.

5.1 Case-base construction

We defined a case as an ordered pair $(\rho, \mu(\rho))$ in which ρ is the problem space, and $\mu(\rho)$ the solution space associated to ρ. In our approach, the problem space ρ is represented by a set of meta-features of the dataset *ds*, attributes, and its target variables, and the solution space $\mu(\rho)$ represents the algorithms used to clean *ds*.

Additionally, we harness the capabilities of Data cleaning ontology (presented in Chapter 4). The cases were represented through Data cleaning ontology, which enhances the integration between cases and domain knowledge [218]. Figure 5.2 shows the representation of the cases in Data cleaning ontology.

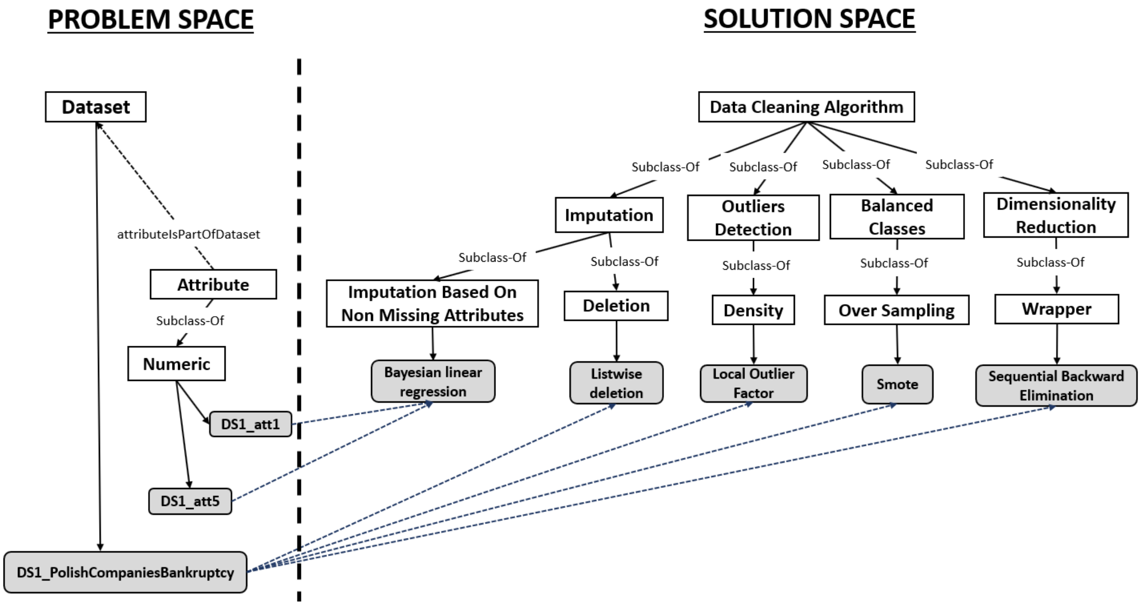


Figure 5.2: Example of case representation through Data cleaning ontology for the dataset of Polish companies bankruptcy. The gray square represents the class individuals while the white square depicts the classes. The solid line means a hierarchical relation and the dotted line indicates the data cleaning algorithms used in the dataset and attributes.

In Figure 5.2 we present an example for the dataset of Polish Companies Bankruptcy [219]. The instances of the dataset: *DS1_PolishCompaniesBankruptcy* and attribute: *DS1_att1*, *DS1_att5* represent the description or problem part of a case, while Data cleaning algorithm instances indicate the solution of the case (*Local Outlier Factor*, *Smote*, *Sequential Backward Elimination*, *ListWise Deletion* and *Bayesian linear regression*).

We collected the datasets from UCI Repository of Machine Learning Databases [158] of the last twenty years (1998 – 2018) based on study of the section 3.5. Thus, we built two case-bases. The first case-base contains 36 cases related with classification datasets (27 datasets presented in section 3.5, and 9 datasets of physical activity monitoring [159]), and the second case-base contains 20 cases for regression datasets.

5.1.1 Problem space

The problem space is described by dataset meta-features. To select the meta-features, we reviewed and analyzed several works focused in meta-learning [220, 221, 222, 223, 224, 225]. Table 5.1 presents a summary of the meta-features found in the meta-learning works.

Table 5.1: Related works of dataset meta-features. The first column presents the research works, follow by the application area and the meta-features.

Works	Area	Meta-features
[220, 221, 222]	Recommendation of feature selection algorithms / Literature review	Number of instances, attributes, dataset dimensionality, mean of absolute linear correlation, skewness, and kurtosis, normalized class entropy, mean normalized feature entropy, mean mutual information of class, mutual information of class, equivalent number of features, noise signal ratio, parameters of models: decision tree, multi layer perceptron, k nearest neighbor.
[223, 224]	Classifier selection	Number of instances, attributes (numeric and nominal), dataset dimensionality , average of entropy, and mutual information, noise signal ratio, interpretability of the model, training and testing time.
[225]	Meta-features on attributes of the dataset	Skewness, kurtosis, entropy, mutual information.

The authors of [220, 221] proposed meta-features of the dataset and model parameters for recommendation of feature selection algorithms. In case of [222], the authors presented a characterization of dataset meta-features, through a literature review of the most frequently used meta-features. Similarly, the works [223, 224] presented approaches for classifier selection based on meta-features of the dataset and model parameters. In [225] with the aim to preserve more information, the authors computed meta-features on attributes of the dataset.

Based on the works mentioned above, we used dataset meta-features and attribute meta-features. Twelve meta-features describe the dataset, and eight meta-features represent each attribute of the dataset (numeric or nominal respectively). Following we expose the first twelve.

- **Instances:** it represents the total number of s samples in the dataset [221].
- **Attributes:** it represents the total number of a attributes in the dataset [221].
- **Data dimensionality:** it is defined as the ratio between the number of attributes a and the number of samples s of the dataset [221]

$$Dim_{data} = \frac{a}{s}$$

- **Missing values ratio:** it represents the mv missing values between the total of $(a \times s)$ values that contain a dataset [224, 223].

$$MissingValues_{ratio} = \frac{mv}{(a \times s)}$$

- **Duplicate instances ratio:** it represents the number of duplicate samples ds between the total of s samples in the dataset [224, 223].

$$DuplicateInstances_{ratio} = \frac{ds}{s}$$

- **Mean absolute linear correlation:** defined as the absolute average of correlations between all the m attributes and the target variable [222].
- **Equivalent number of features:** it indicates if the number of attributes in a given dataset is suitable to optimally solve a classification task (under the assumption of independence among attributes). This is expressed as the number of attributes would be required, on average, by taking the ratio between the entropy target variable (nominal) $H(Class)$ and the average mutual information $\overline{MI(Class, nomAtt)}$ [222].

$$EN_{att} = \frac{H(Class)}{\overline{MI(Class, nomAtt)}}$$

- **Mean absolute skewness:** defined as the absolute average of skewness over all the m numerical attributes [225].
- **Mean absolute kurtosis:** defined as the absolute average of kurtosis over all the m numerical attributes [225].
- **Mean attribute entropy:** defined as the average of normalized entropy over all the m nominal attributes [225].
- **Mean mutual information:** defined as the average of mutual information between all the m nominal attributes and target variable [225].

- **Noise-signal ratio:** it represents the amount of irrelevant information contained in a dataset [222]. The expression used to evaluate this feature is

$$NS_{ratio} = \frac{\overline{H(nomAtt)} - \overline{MI(Class, nomAtt)}}{\overline{MI(Class, nomAtt)}}$$

Following we expose the eight meta–features that describe the attributes of the dataset. The missing values and correlations coefficient correspond to numeric and nominal attributes.

- **Missing values:** defined as ratio of missing values $att_{missingValues}$ of the total values att_{values} of an attribute [224, 223].

$$MissValAtt_{ratio} = \frac{att_{missingValues}}{att_{values}}$$

- **Correlation:** attempts to measure the strength of a relationship between the target variable and an attribute [224, 223]. The correlation coefficient can take values in the interval $[-1, 1]$, where 1 indicates a strong positive relationship and -1 a strong negative relationship. A result of 0 indicates no relationship at all.

When the attribute is numeric, three features are computed:

- **Candidate outliers:** we calculated the candidate outliers based on Tukey Fences [226]. Values of a numeric attribute below $Q_1 - 1.5(Q_3 - Q_1)$ or above $Q_3 + 1.5(Q_3 - Q_1)$ are considered potential outliers [161], where Q_1 and Q_3 are the first and third quartile respectively. Once detected the candidate outliers, we calculated outliers ratio defined as candidate outliers between total values $numAtt_{values}$ of a numeric attribute.

$$Outliers_{ratio} = \frac{candidateOutliers}{numAtt_{values}}$$

- **Kurtosis:** it measures the peakedness in the distribution of a numeric attribute $numAtt$. Positive values indicate a higher, sharper peak (leptokurtic). Negative value mean a lower, less distinct peak (platykurtic). A normal distribution has kurtosis close to zero (mesokurtic). The kurtosis is represented by the ratio of the fourth moment of the distribution of a numeric attribute $numAtt$ to the fourth power of the standard deviation [225]

$$Kurt_{numAtt} = \frac{1}{std^4_{numAtt}} \frac{\sum_{k=1}^{n}(numAtt_k - \overline{numAtt})^4}{k}$$

- **Skewness:** it indicates the lack of symmetry in the distribution of a numeric attribute $numAtt$. Negative skewness values indicate data that are skewed left, while positive skewness values denote data that are skewed right. In case of zero value, the distribution is symmetric. The skewness is represented by the third moment of the distribution of a numeric attribute $numAtt$, divided by the third power of standard deviation [225]

$$Skew_{numAtt} = \frac{1}{std^3_{numAtt}} \frac{\sum_{k=1}^{n}(numAtt_k - \overline{numAtt})^3}{k}$$

In case of nominal attributes three features are defined:

- **Normalized Entropy:** indicates the degree of uniformity of the distribution of a nominal attribute $nomAtt$ [222]. Denoted by

$$H(nomAtt) = -\sum_{i=1}^{n} q_i log_2(q_i)$$

Where $q_i = p(nomAtt = x_i)$ is the probability that $nomAtt$ assumes the ith value x_i, for $i = 1, ..., n$. We suppose that each value of a nominal attribute in a dataset has the same probability of appearing, therefore the theoretical maximum value for the entropy of the nominal attribute is $log_2(n)$. Thus the normalized entropy can be computed as:

$$H(nomAtt)_{norm} = -\sum_{i=1}^{n} \frac{q_i log_2(q_i)}{log_2(n)}$$

- **Mutual information:** measures the common information shared between the target variable (nominal) C and nominal attribute $nomAtt$ [222]. The mutual information of a class and an attribute is defined as:

$$MI(Class, nomAtt) = H(Class) + H(nomAtt)?\,H(Class, nomAtt)$$

- **Labels:** corresponds to the number of values of the nominal attribute.

We use the same features of an attribute for a target variable (numerical or nominal). When the target variable is nominal, additionally we use the imbalance ratio to measure the distribution of the classes:

$$IR = \frac{Class^+}{Class^-}$$

$Class^+$ represents the size of the majority class and $Class^-$ the size of the minority class. A dataset with IR 1 is perfectly balanced, while datasets with a higher IR are more imbalanced [156].

5.1.2 Solution space

The solution space is represented by the algorithms and parameters (of the data cleaning tasks: imputation, outlier detection, classes balancing, label correction, remove duplicate instances and dimensionality reduction) used for cleaning each dataset.

As mentioned in subsection 3.5, we use 56 datasets from UCI Repository of Machine Learning Databases [158] (36 cases for classification and 20 for regression tasks). Each one of these datasets has publications of the results of classification or regression models used. To guarantee a correct space solution, we preprocessing all 56 datasets using our conceptual framework for data cleaning in knowledge discovery tasks presented in Chapter 3. We described in subsection 3.5 that the results achieved by the trained models with the dataset produced by our conceptual framework reached high or similar performance compared with the models presented by the authors of UCI datasets.

The low number of cases of our case-base is due to availability data of the domain and restrictions for dataset selection (each one of the selected datasets must have publications). For example, in similar domains, the CBR for selection of classification and regression models [102], the case-base contains 80 cases. Others domains as authors of [87] where CBR is built for the diagnosis of gastrointestinal, the case-base contains 53 cases. The CRB proposed in [98] for construction cost of multi-family housing complexes, the case-base is composed by 99 cases, while the CBR for web service discovery and selection developed in [96] built a case-base of 62 cases.

5.2 Case retrieval

As mentioned in subsection 2.1.4, the common retrieval mechanisms of a CBR are based on similarity measures and filtering methods. Thus, we propose a case retrieval mechanism composed of a filter and similarity phases. In the first phase, we defined two filter approaches based on clustering and quartile methods. These filters retrieve a reduced number of relevant cases. The second phase computes a ranking of recovered cases by filter approaches and generates similarity scores between the new case and the retrieved cases. In the second phase, we proposed two similarity mechanisms based on meta-features of dataset and attributes. Figure 5.3 presents the case retrieval architecture.

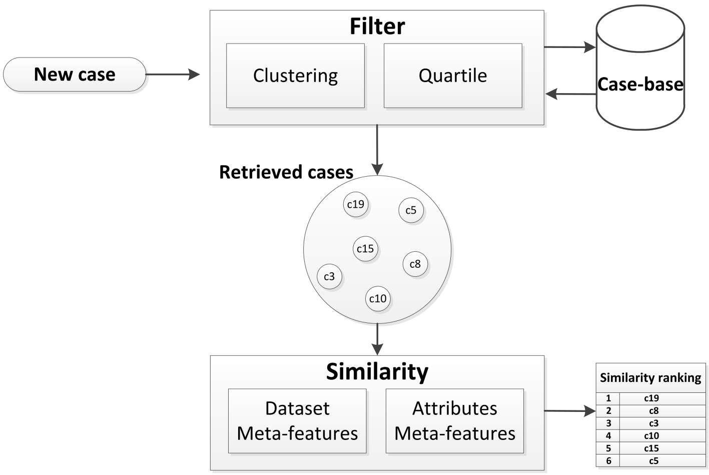


Figure 5.3: Case retrieval mechanism. In the first phase, a filter approach is applied. The second phase computes a ranking of recovered cases by filter approaches based on similarity measures.

5.2.1 Filter phase

This phase retrieves the relevant cases respect to the new case. We propose two filter methods. These filters are presented below:

5.2.1.1 Case clustering

The purpose of this method is group cases into subsets called clusters [227, 228]. Thus, given a new case C_q, this one is assigned in a $Cluster_i$ when it has a high degree of similarity respect the case stored into $Cluster_i$. We used *k-means* as cluster algorithm, a popular partition method widely used in the data mining community [229, 230].

Before classifying a new case C_q, we must define the number of clusters (for classification and regression tasks). As mentioned earlier, we use the *k-means* algorithm for this process. First, *k-means* randomly selects k cases from the whole case-base. These cases represent the initial centroids (or seeds). Each remaining case of the case-base is assigned to a cluster whose centroid is the closest to that case. The coordinates of the centroid are then recalculated. The new coordinates of a specific centroid correspond to the average of all cases assigned to the respective cluster. This process iterates until a cost function converges to an optimum without a guarantee that it is the global one [231]. Figure 5.4 presents an example of K-means with 3 centroids applied on 12 cases.

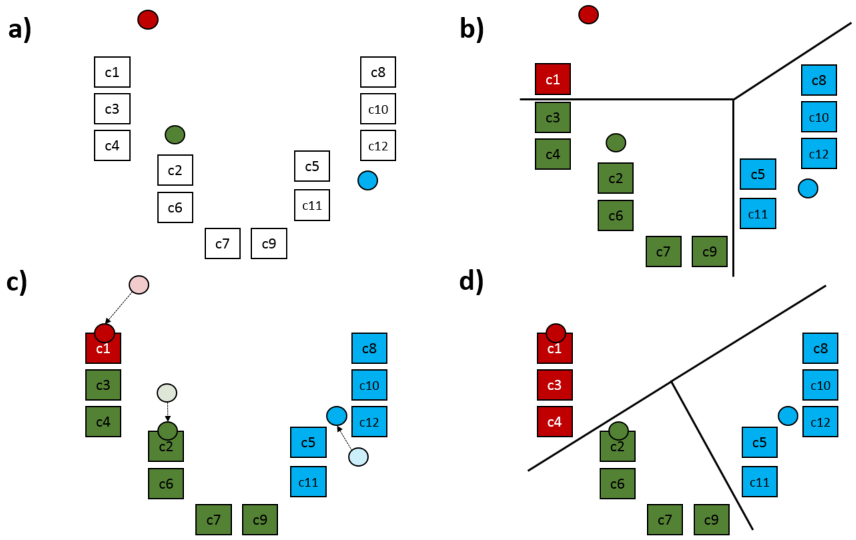
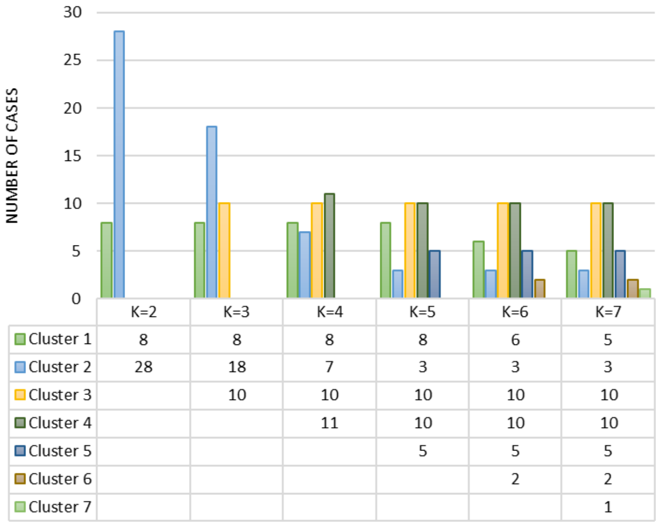


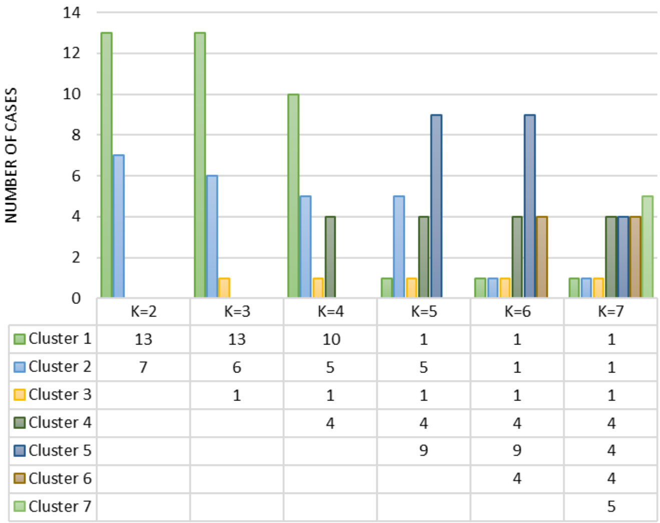
Figure 5.4: Example of K-means. The circles represent the centroids and the squares depict the cases. a) The initial centroids are assigned. b) The cases are assigned to the closet cluster centroids. c) The cluster centroids are recalculated. d) The cases are assigned to new centroids.

We tested the space problems of the cases with k-means with 2, 3, 4, 5, 6 and 7 clusters, for classification and regression cases. Figures 5.5 and 5.6 present the cases distribution in the clusters for classification and regression case-bases.



	K=2	K=3	K=4	K=5	K=6	K=7
Cluster 1	8	8	8	8	6	5
Cluster 2	28	18	7	3	3	3
Cluster 3		10	10	10	10	10
Cluster 4			11	10	10	10
Cluster 5				5	5	5
Cluster 6					2	2
Cluster 7						1

Figure 5.5: Case distribution in the clusters for case-base of classification. The x-axis corresponds to the number of clusters and y-axis number represents cases assigned to each cluster



	K=2	K=3	K=4	K=5	K=6	K=7
Cluster 1	13	13	10	1	1	1
Cluster 2	7	6	5	5	1	1
Cluster 3		1	1	1	1	1
Cluster 4			4	4	4	4
Cluster 5				9	9	4
Cluster 6					4	4
Cluster 7						5

Figure 5.6: Case distribution in the clusters for case-base of regression. The x-axis corresponds to the number of clusters and y-axis number represents cases assigned to each cluster.

To classify a new case C_q in a specific cluster, we built a decision tree from C4.5, Multilayer Perceptron (MLP) and Support Vector Machine (SVM) from Weka tool kit for 2, 3, 4, 5, 6 and 7 clusters. We used the default experimental configuration of Weka to build the classifiers. As validation method we used cross validation with 10 folds.

In this case, we are interested to assess the proportion of cases that belong correctly to a cluster, due to we must guarantee to user the most similar cases respect to new case. Thus, we used the True Positive (TP) Rate as performance measure. Figures 5.7 and 5.8 present the True Positive (TP) Rate for the obtained models.

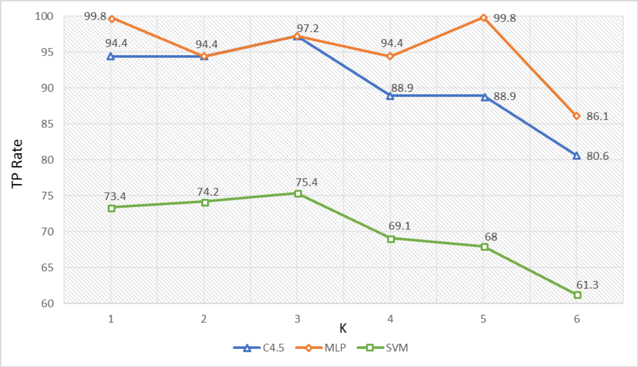


Figure 5.7: True Positive Rate of C4.5, MLP and SVM (2, 3, 4, 5, 6 and 7 clusters) for case-base of classification.

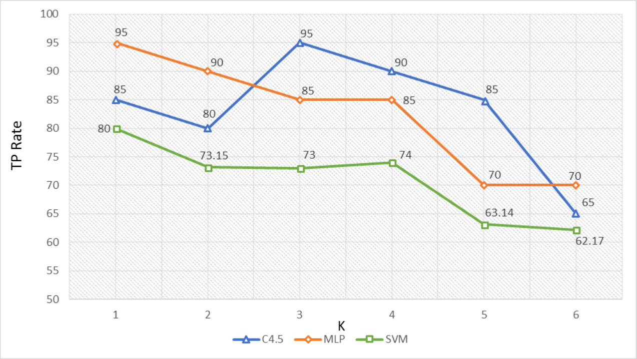


Figure 5.8: True Positive Rate of C4.5, MLP and SVM (2, 3, 4, 5, 6 and 7 clusters) for case-base of regression.

We selected the models with highest true positive (TP) rate for classification and regression tasks (Figures 5.7 and 5.8). MLP with 6 clusters was the model with highest TP rate for classification tasks (99.8%), whereas, in regression tasks, C4.5 with 4 clusters achieves the highest TP rate (95%).

5.2.1.2 Case quartile

Quartiles capture fundamental information about a variable distribution that complements other traditional metrics like the mean, mode, and standard deviation [232]. For calculate the quartile, first the variable must be arranged in ascending order; subsequently it is divided into four equal parts Q_1, Q_2, Q_3, Q_4 quartile [233, 234].

- First quartile Q_1 means that about 25% of the values in the variable lie below Q_1.

- The second quartile Q_2 (or median) cuts the values of the variable in half.
- The third quartile Q_3 means that about 75% of the values in the data set lie below Q_3 and about 25% lie above Q_3.
- The fourth quartile Q_4 corresponds to maximum value of the variable.

In this approach, we apply the quartile analysis to the features of dataset defined in subsection 5.1. Figure 5.9 shows an example of quartile analysis for 12 cases arranged by missing values ratio, mean absolute kurtosis and mean attribute entropy.

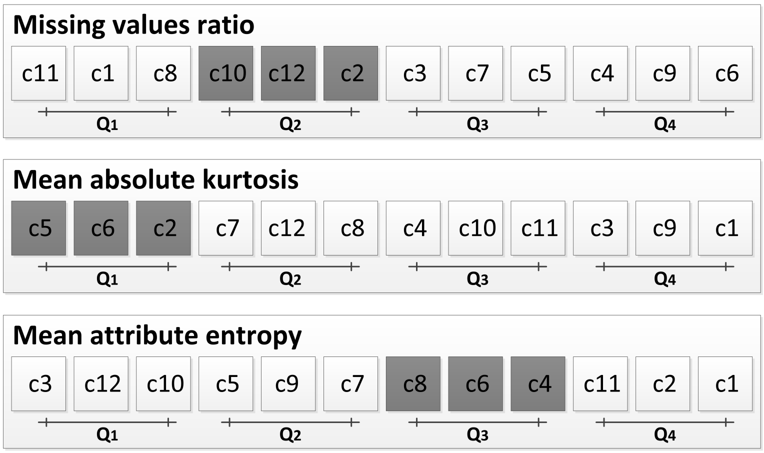

Missing values ratio
c11 c1 c8 c10 c12 c2 c3 c7 c5 c4 c9 c6
Q1 Q2 Q3 Q4
Mean absolute kurtosis
c5 c6 c2 c7 c12 c8 c4 c10 c11 c3 c9 c1
Q1 Q2 Q3 Q4
Mean attribute entropy
c3 c12 c10 c5 c9 c7 c8 c6 c4 c11 c2 c1
Q1 Q2 Q3 Q4

Figure 5.9: Example of quartile analysis for missing values ratio, mean absolute kurtosis and mean attribute entropy. The gray cells correspond to quartiles where the new case C_q is classified.

Thus, a new case C_q is classified in a quartile according to values of the dataset features. In the example of Figure 5.9, C_q is classified in Q_2 of missing values ratio, Q_1 of mean absolute kurtosis and Q_3 of mean attribute entropy. Finally, the cases C_{10}, C_{12}, C_2, C_5, C_6, C_8, C_4 of the quartiles Q_2, Q_1 and Q_3 (omitting the duplicate cases) are the most similar cases respect to C_q.

With the aim to select the best filter mechanism to reduce the search space, in Section 5.5, we present the evaluation of the two filter approaches for classification and regression tasks respectively.

5.2.2 Similarity mechanisms

The purpose of these mechanisms is to find the most similar case of the case-base given new case through similarity measures. When two cases are compared, the similarity measures are applied to dataset meta-features that describe the problem space of a case.

These mechanisms compute a similarity ranking of the retrieved cases by filter approaches. We proposed two similarity mechanisms, the first one based on dataset meta-features, and the second one in meta-features of dataset attributes.

5.2.2.1 Similarity based on dataset meta-features - Sim(ds)

The attribute-value representation of a case is defined as vector of dataset meta-features (Subsection 5.1): $C_i = [metFeat_1, metFeat_2, ..., metFeat_n]$ where i represents the ith case. Therefore, the assessment of similarity between two cases C_q and C_t is given by:

1. The similarity between values of attributes (local similarities):

   $$Sim_{metFeat_j}(C_q(metFeat_j), C_t(metFeat_j))$$

   Where C_q is the query case, C_t the target case, and j the jth feature.

2. The global similarity between C_q and C_t cases. This measure consists of a sum of local similarity measures and assumes a limited value between 0 and 1:

   $$\sum_{j=1}^{n} W_j * Sim_{metFeat_j}(C_q(metFeat_j), C_t(metFeat_j))$$

   Where W_j is the weight of the jth feature.

In case of the normalized features of the dataset (Missing values and Duplicate instances ratio, mean absolute linear correlation and mean attribute entropy), we use the weighted Euclidean measure [235] as local similarity function:

$$1 - \sqrt{\sum_{j=1}^{n} W_j * (C_q(metFeat_j) - C_t(metFeat_j))^2}$$

For the non–normalized features of the dataset with high dispersion, as instances, attributes, data dimensionality and mean absolute skewness, the equivalent number of features and noise-signal ratio, we use the weighted Canberra similarity [236], due this measure is sensitive to proportional differences and it allows to identify deviations from normal observations. The weighted Canberra is defined:

$$1 - \sum_{j=1}^{n} W_j * \frac{|C_q(metFeat_j) - C_t(metFeat_j)|}{|C_q(metFeat_j)|+|C_t(metFeat_j)|}$$

For the remaining of non-normalized features (mean absolute kurtosis and mean mutual information), where the standard deviations are low, we used the arithmetic summation-based similarity:

$$1 - \sum_{j=1}^{n} W_j * \frac{|C_q(metFeat_j) - C_t(metFeat_j)|}{Max(C_t(metFeat_j)) - Min(C_t(metFeat_j))}$$

Where $Max(C_t(metFeat_j)) \neq Min(C_t(metFeat_j))$

5.2.2.2 Similarity based on meta-features of dataset attributes - Sim(att)

In this subsection, we explain the second similarity mechanism proposed.

We built an attribute-value approach working on meta-features of the attributes and target variable of a dataset (Section 5.1). The case of attribute-value approach is represented by a vector of dataset attributes and target variable $C_i = [numAtt_1, ..., numAtt_n, nomAtt_1, ..., nomAtt_n, target]$.

The numeric attribute $numAtt$ represents the set of features: *outliers*, *kurtosis*, and *skewness*; while the attribute $nomAtt$ represents the features: *entropy*, *mutual information*, and *labels*. Additionally, the numeric or nominal attributes share the features: *missing values* and *correlation*.

In case of $target$, the numeric variable is represented by three features: *outliers*, *kurtosis*, and *skewness*, while nominal target variable by two features: *entropy* and *labels*.

This attribute-value approach was implemented using a Global Similarity Function (GSF), it integrates the similarity measures of numeric and nominal attributes, and the target variable:

$$\beta_1 simNumAtt(C_q, C_t) + \beta_2 simNomAtt(C_q, C_t) + \\ \rho simTarget(C_q, C_t)$$

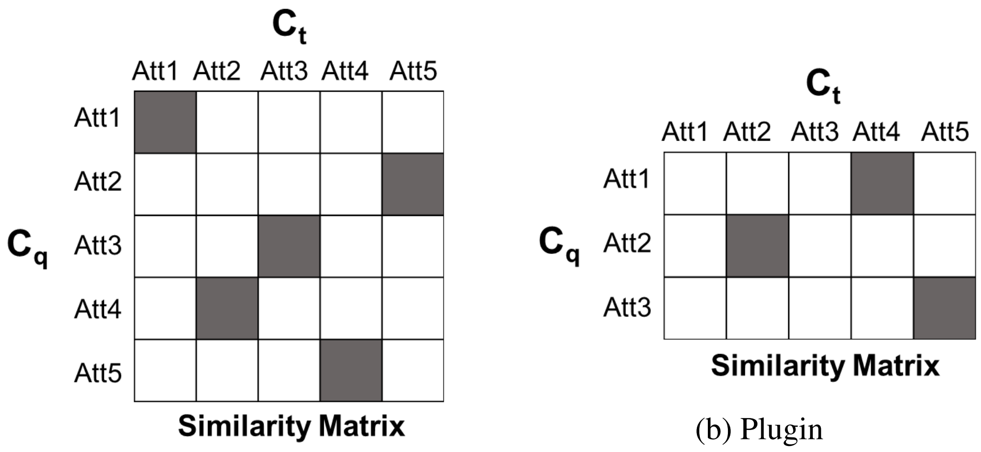
Where β_1, β_2, and ρ represents the weights of each similarity function. Below we explain how we calculate the similarity measures of attributes and target variable:

Similarity between attributes

First, we compared the number of numeric and nominal attributes of C_q and C_t through attribute matching:

- Exact: the number of attributes (between numeric or nominal) of C_q is equal to number of attributes of C_t (Figure 5.10a).
- Plugin: the number of attributes (between numeric or nominal) of C_q is less than number of attributes of C_t (Figure 5.10b).
- Subsume: the number of attributes (between numeric or nominal) of C_q is greater than number of attributes of C_t (Figure 5.11).

Once defined the attribute matching, we computed the similarity for each attribute (between numeric or nominal) of C_q against all attributes of C_t, then the results are stored in a similarity matrix. Subsequently, we selected the highest similarity obtained by each attribute of C_q respect to C_t attributes, where each attribute of C_t must be different for each attribute of C_q. Figure 5.10 presents an example of attribute matching for Exact and Plugin categories, where the gray cells represent the highest similarity for each attribute of C_q respect to C_t attributes.



(a) Exact

(b) Plugin

Figure 5.10: Attribute matching for Exact and Plugin categories. The first matrix shows the Exact attribute matching. The second column presents the Pluging matching. The rows represent the dataset attributes of C_q, while the columns depict the dataset attributes of C_t. The gray cells represent the highest similarity for each attribute of C_q respect to C_t attributes

In case of Subsume attribute matching (Figure 5.11), due the number of attributes of C_q is greater than the number of attributes of C_t, an attribute of C_t can be used several times to calculated the similarity between C_q attributes. Therefore, we calculated the transpose of the similarity matrix, after we selected the highest similarity obtained by each attribute of C_t respect to C_q attributes. Also, we defined a penalization $\alpha = da/C_q(atts)$, where da is the number of discarded attributes of C_q for computing similarities and $C_q(atts)$ the attributes of C_q. Figure 5.11 presents an example of Subsume attribute matching, where C_q and C_t have 3 and 5 attributes respectively, with a penalization $\alpha = 0.4$.

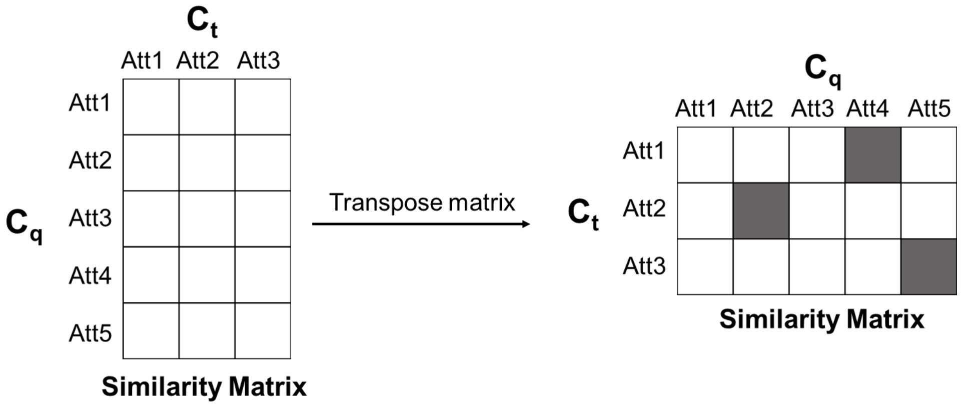


Figure 5.11: Subsume match of dataset attributes. The rows of the first matrix represent the dataset attributes of C_q, while the columns of the first matrix depict the dataset attributes of C_t. The second matrix is the transpose of similarity matrix. The gray cells represent the highest similarity for each attribute of C_q respect to C_t attributes

Finally, the highest similarities of numeric and nominal attributes are averaged.

Similarity between target variables

We calculate with local similarity functions the numeric feature set (*outliers*, *kurtosis*, *skewness*) and nominal (*entropy* and *labels*) target variables of C_q and C_t.

We used as local similarity functions the Euclidean, Canberra and Arithmetic distance. The step–by–step to calculate the similarity between target variables is: If the feature is normalized, we used Euclidean distance. If the feature is not normalized and it has high dispersion, the Canberra distance is used, in otherwise we used Arithmetic distance. Table 5.2 presents the similarity functions used in the features of attributes and target variable.

Table 5.2: Similarity functions used in features of attributes and target variable

Variable	Feature	Similarity function
Attribute	Correlation	Arithmetic
	Missing values	Euclidean
Numeric attribute	Candidate outliers	Euclidean
	Kurtosis	Canberra
	Skewness	Canberra
Nominal attribute	Normalized Entropy	Euclidean
	Mutual information	Arithmetic
	Labels	Canberra
Target variable	Missing values	Euclidean
Numeric target variable	Candidate outliers	Euclidean
	Kurtosis	Canberra
	Skewness	Canberra
Nominal target variable	Normalized Entropy	Euclidean
	Labels	Canberra

In section 5.5, we present the results of the filter approaches and similarity mechanisms.

5.3 Case reuse

Giving a retrieved case C_t, the system adjust the solution space (data cleaning algorithms) of C_t as a solution of C_q [85]. If the problem space of case C_q is precisely like to C_t (which is supposed to have been successful), then copy the old data cleaning solution [48]. In the event of problem space of C_q is different to C_t, the recommendation is to adapt the recorded data cleaning solution before reusing it, to ensure the best suit the new data quality issues [237].
In addition, the ontology proposed in Chapter 4 plays a key role in reuse phase. This recommends similar data cleaning algorithms to the algorithm proposed in the case solution of C_t. For example, Figure 5.12 depicts the taxonomy of dimensionality reduction algorithms of the Data cleaning ontology (DCO). The individuals of DCO: information gain, gain ratio, Pearson correlation, symmetrical uncertainty or chi-squared correspond to dimensionality reduction algorithms based on filter approach. Assuming the solution of the retrieved case C_t was information gain (filter approach), the Data cleaning ontology presents to the user, similar

filter algorithms as gain ratio, Pearson correlation, symmetrical uncertainty or chi-Squared.

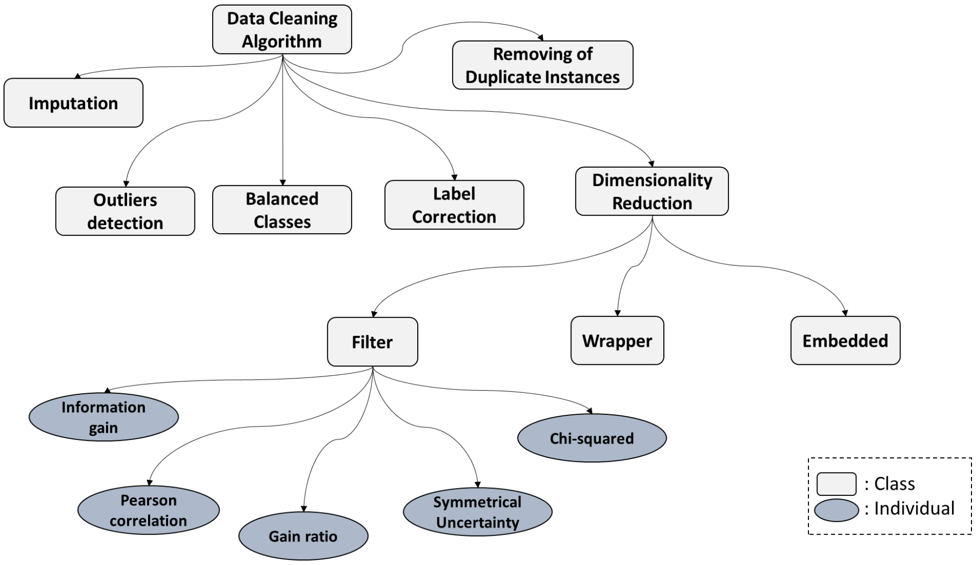


Figure 5.12: Representation of dimensionality reduction algorithms in data cleaning ontology. The blue circle represents the class individuals while the gray square depicts the classes. The solid line means a hierarchical relation.

5.4 Case retain

The retain step stores the case C_q (dataset meta–features and data cleaning solution) into the temporal case-base for future reuse. The solution of the adapted case must be tested before save it in the case-base. We reviewed approaches for the evaluation of adapted cases [48]:

1. **Human experts** that review the validity of data cleaning methods applied. The disadvantages are the availability and the susceptibility to errors of the experts. These problems can be improved if the experts are replaced by a documented formal process.

2. **Evaluate in the real world** the solution of the adapted case. The results of the application the data cleaning algorithms in classification and regression tasks can notify us feedback from reality.

Although the evaluation of adapted cases by human experts is a complex process due to the verification of each new case takes a long time (we must prevent bad solutions being retained) [86], we consider it, the best evaluation approach than evaluation in the real world because the second approach evaluates the adapted case after to the application in the real world. Therefore, we propose to verify the quality of the C_q case through human experts supported in three data quality dimensions [7, 238]:

1. Completeness verifies the case was having all required parts (data quality issues and data cleaning solutions) [239].
2. Validity is the degree to which the case conforms to a set of rules, represented within a defined data domain (e.g., if a dataset does not contain missing values, then the imputation algorithms do not use) [7].
3. Accuracy refers when the data cleaning algorithms of the case solution were applied to dataset and the model generated by the cleaned dataset obtains good results [240, 241]. The measurement of accuracy depends highly of the experts. They must verify the performance of the models based on statistical measures, their knowledge, and the domain [48].

5.5 Results

As mentioned in Subsection 2.1.4, a CBR is essentially centered on retrieval mechanism of cases [48]. The case retrieval is considered a key phase in CBR, due it establishes the foundation for the general performance of CBR systems [242]. The aim of the retrieval mechanisms is to retrieve the most similar case that can be successfully used to solve a new problem. If the retrieval mechanism fails, CBR system will not produce good solutions for the new problem [243]. Thus, we focus on the evaluation of the case retrieval mechanism proposed in Section 5.2. We used a Collaborative Evaluation Methodology [244] which is composed of two steps: judges evaluation, and review of judges evaluation.

In the first step, a panel of judges assess the retrieval mechanism. Table 5.3 presents the panel of judges and their experience in data mining projects.

Table 5.3: Experience in data mining projects of the judges

Judge Id	Data mining experience	Years of experience
1	Master and Phd thesis	5 years
2	Teacher	3 years
3	Phd thesis	3 years
4	Phd thesis	2.5 years
5	Master thesis	2 years
6	Master thesis	2 years

The panel of judges scores (0-100%) the similarity between a query case against all cases of the case-base. For each query case, a list of case similarity is returned. We defined three kind of queries for each knowledge discovery tasks (classification and regression):

1. Query 1: corresponds to a copy of a case contained in the case-base. This query verifies the minimum quality of the retrieval mechanism. The retrieval mechanism and panel of judges should obtain 100% of similarity for Query 1 respect to identical case contained in the case-base.
2. Query 2: is a modified case of a case contained in the case-base. The retrieval mechanism and panel of judges should obtain a high similarity between Query 2 and the case non-modified of the case-base.
3. Query 3: is a new case, it is not contained in the case-base. The aim of this query is to simulate the behavior of the retrieval mechanism in the real world.

The results considered relevant by the panel of judges will be those that represent the ideal responses for each query case. The evaluation in detail of judges for each query case is presented in Appendix B.1.

In the second step, we reviewed one by one the relevance judgments issued in the previous stage, and we compared the judgments that other judges have stated. If a judge evaluation is a discordant respect to the other judges evaluations, we discarded the discordant evaluation. In our experiment, the evaluations of the judge 4 were discarded due the assessment of the cases are very low compared with the remaining of the judges. Subsequently, the selected evaluations are averaged and we generated ranking of cases proposed by the panel of judges.

Finally, the ranking of cases proposed by the panel of judges is compared with the ranking of cases obtained by our case retrieval mechanism. To evaluate the quality of the ranking generated by our retrieval mechanism, we used two measures of retrieval information [245, 246, 247]:

- *Precision@K:* proportion of retrieved cases that are relevant in the judges ranking of K positions:

  $$P@K = \frac{Rel_{cases}}{K}$$

  Where Rel_{cases} is the number of relevant cases and K the ranking size.

- *P–Precision@K:* proportion of relevant retrieved cases in the same positions of the ranking Top–K of the judges:

$$P - Precision@K = \frac{P - Rel_{cases}}{K}$$

  Where $P - Rel_{cases}$ is the number of relevant cases located in the same positions of the judges ranking and K the ranking size.

5.5.1 Classification

To assess the retrieval mechanism for classification tasks, we selected randomly from case-base the first two queries. The third query was took from UCI repository. The dataset of query 3 was created in 1996 (out of years range of the collected datasets for building the case-base). Below the case queries are presented:

- *Query 1 – Autism spectrum disorder in children* is a copy of a case contained in the case-based and it describes a children screening data for autism spectrum disorder [166].

- *Query 2 – Portuguese bank telemarketing (modified)* is a modified case of the case-base. We deleted three attributes and 39.000 instances. This query is related with direct marketing campaigns (phone calls) of a Portuguese banking institution [177].

- *Query 3 – Income prediction* corresponds a new case. This query represents the income of a person in United States exceeds 50.000 USD per year based on census data [248].

For each query case, we applied the filter approaches (Clustering and Quartile) to obtain the most similar cases. Figure 5.13 presents the number of retrieved cases by filter approach.

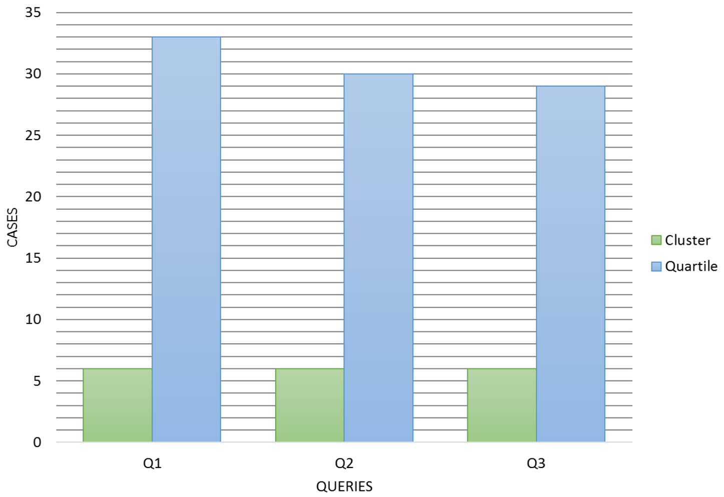


Figure 5.13: Retrieved cases by filter approach for classification tasks

In Figure 5.13, the clustering filter retrieves 5 cases for all queries, while quartile approach 33 cases for Query 1 (Q1), 30 for Query 2 (Q2) and 29 for Query 3 (Q3). In other words, clustering approach is a rigorous filter because it retrieves 13.88% of the cases while quartile approach retrieves more than 80% of the cases which can be irrelevant cases.

To verify the precision of the retrieved cases by filter approaches, in Figure 5.14 we present the Precision@K with *P@3, P@7, and P@10*.

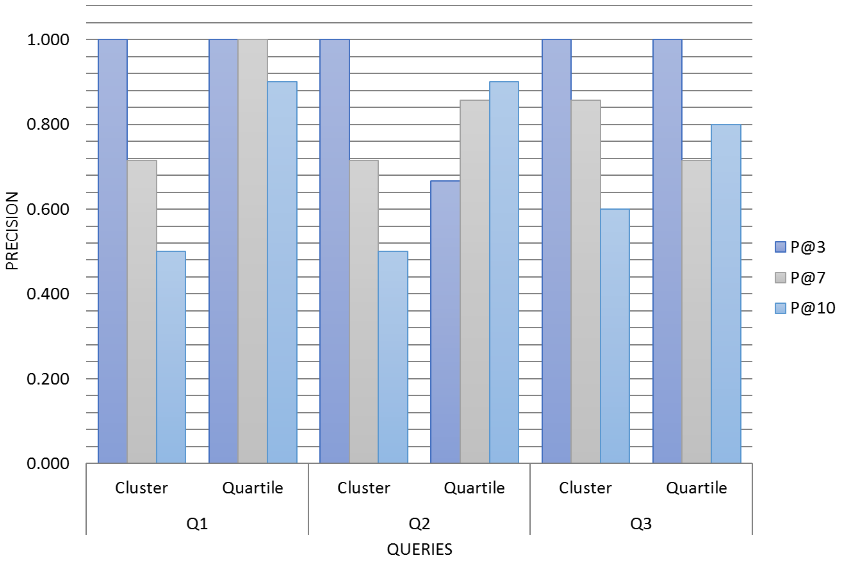


Figure 5.14: Precision *P@3, P@7*, and *P@10* for filter approaches in classification tasks

In case of P@3, the filter approaches retrieve 100% of relevant cases for all queries. Quartile filter reaches the highest precisions for *P@7* in Q1 (100%) and Q2 (85.7%), and clustering filter by Q3 (85.7%). The quartile filter obtains the highest precision in *P@10* for Q1 (90%) and Q2 (90%) and Q3 (80%). The highest precisions were obtained by quartile filter because this approach retrieves a large number of cases compared with clustering filter.

To evaluate the ranking quality of the filter approaches and similarity mechanisms, in Figure 5.15, We show *P-P@1, P-P@2, P-P@3, P-P@4*, and *P-P@5* for Q1, Q2, and Q3.

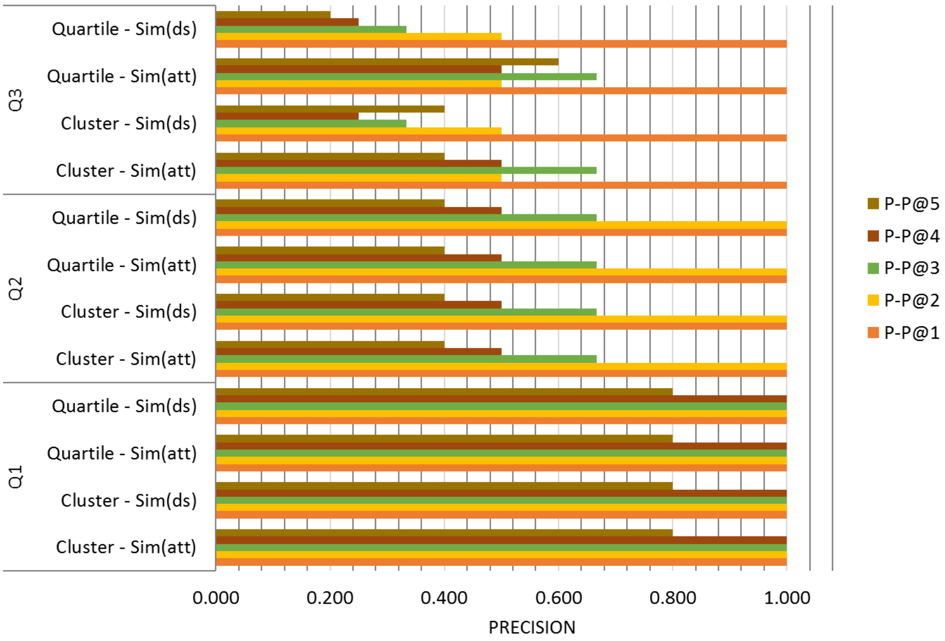


Figure 5.15: Top5 – P–Precision@K for filter approaches and similarity mechanisms in classification tasks

The filters and similarity mechanisms reach 100% of precision in *P-P@1* for all queries, *P-P@2* for Q1, Q2, *P-P@3* and *P-P@4* for Q1. These results mean that our approaches retrieve correctly the first two positions of the judges ranking for queries Q1, Q2, Q3, and the top three and four positions for Q1. In case of *P-P@5* the highest precisions are achieved in Q1 by all approaches (80%), and Q3 by quartile approach using a Sim(att) mechanism (60%).

In general, we consider suitable the clustering filter for classification tasks, due this retrieves 5 cases which 3 cases are relevant in top–3, in contrast to quartile approach, which extracts a large number of irrelevant cases. Respect to similarity mechanisms, they achieve the same precision for Q1 and Q2. However in Q3, Sim(att) obtains highest precisions in *P-P@3, P-P@4* and *P-P@5*, this means that Sim(att) is closer to the judge rating than Sim(ds).

5.5.2 Regression

To assess the retrieval mechanism for regression tasks, we selected randomly from case-base the first two queries. The third query was selected based on previous work developed ourselves in coffee rust. Below the case queries are presented:

- *Query 1 – Air pollution – benzene estimation* is a case of the case-base. This contains information of a gas multi-sensor device deployed on the field in an Italian city [204, 249].
- *Query 2 – Rental bikes hourly* is a modified case of the case-base. We deleted one attribute and 8.500 instances. This query contains the hourly count of rental bikes between years 2011 and 2012 in Capital bikeshare system [203].
- *Query 3 – Coffee rust* is a new case, it is not included in the case-base. This query addresses coffee rust detection in Colombian crops [135, 134, 250].

Figure 5.16 presents the number of retrieved cases by filter approaches in the case-base of regression tasks.

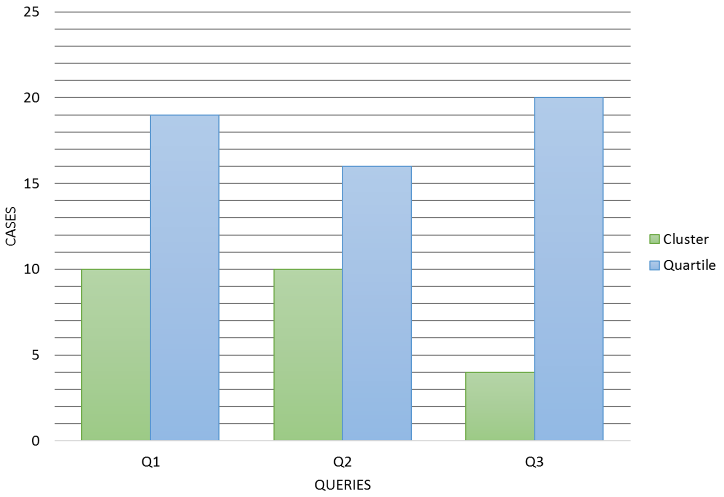


Figure 5.16: Retrieved cases by filter approach for regression tasks

Similar to filters of the classification tasks, the clustering approach retrieves a suitable number of cases compared with quartile approach. The filter clustering

retrieves 10 cases for Q1, Q2, and 4 cases for Q3, while filter clustering retrieves 19 cases for Q1, 16 for Q2 and all cases (20) of the case-base for Q3.
In this sense, in Figure 5.14 we present the precision (*P@3*, *P@7*, and *P@10*) of the retrieved cases by filter approaches.

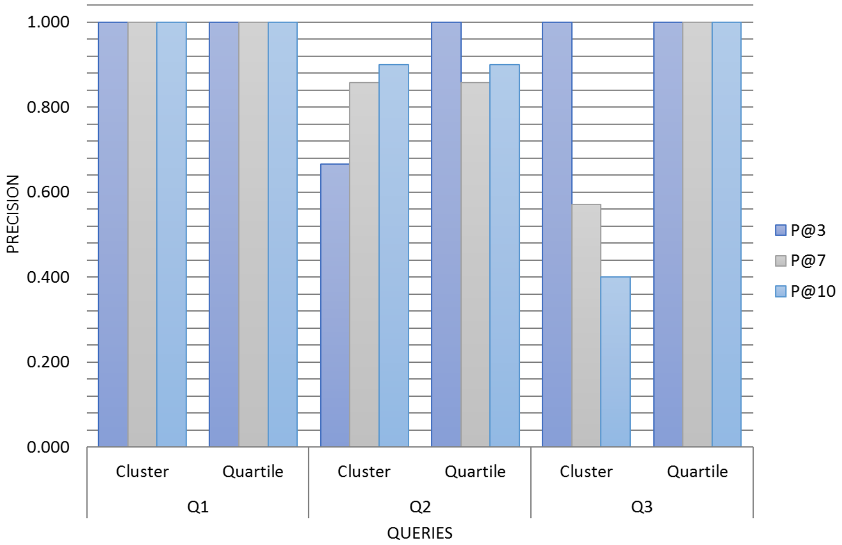


Figure 5.17: Precision *P@3, P@7*, and *P@10* for filter approaches in regression tasks

For Q1, the filter approaches retrieve 100% of relevant cases in *P@3, P@7*, and *P@10*. In Q2, quartile filter achieves the highest precision for *P@3* (100%), while in *P@7* (85.70%) and *P@10* (90%) the filter approaches reach the same precision. For Q3, the quartile filter retrieves 100% of relevant cases in *P@3, P@7*, and *P@10* due this filter retrieves all cases of the case-base.

Finally, to evaluate the ranking quality of the filter approaches and similarity mechanisms in regression tasks, in Figure 5.18, we present *P-Precision@K* measure for top five positions.

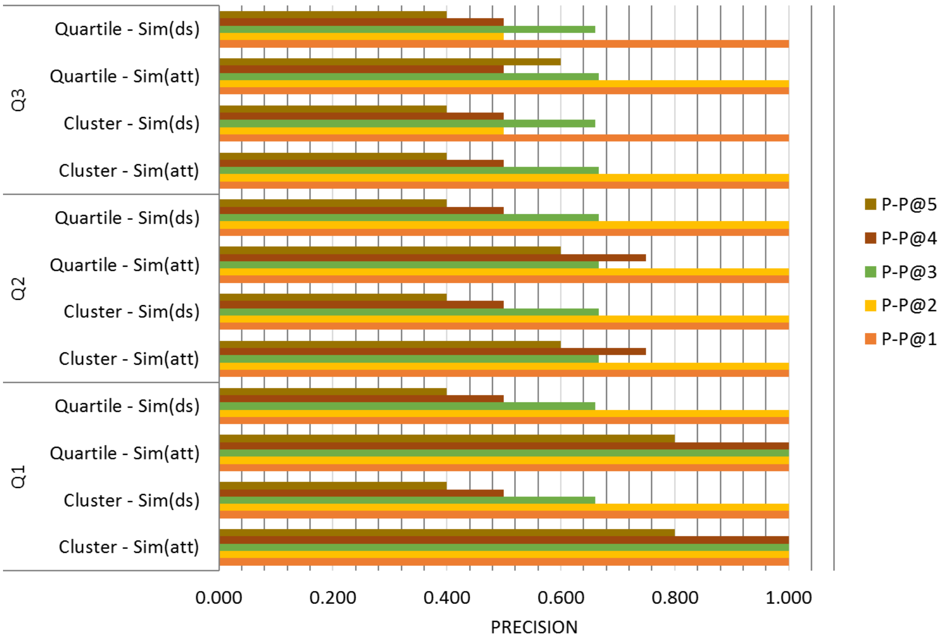


Figure 5.18: Top5 – P–Precision@K for filter approaches and similarity mechanisms in regression tasks

The retrieved cases for the filter approaches and similarity mechanisms of the Q1 show 100% of precision in *P-P@1*, *P-P@2*, *P-P@3*, *P-P@4*, and 80% of precision for *P-P@5*. Likewise, in Q2 all filter approaches and similarity mechanisms for *P-P@1, P-P@2* achieve 100% of precision, while *P-P@3* achieves 66.70% of precision for all approaches. The highest precision in *P-P@4* (75%), and *P-P@5* (60%) are reached by filter approaches using Sim(att). In case of Q3, *P-P@1* reaches 100% of relevant cases for all approaches, while the filters methods using Sim(att) reach 100% of precision in *P-P@2*. The highest precision in *P-P@3* (66.70%) and *P-P@4* (50%) are achieved by all approaches, for *P-P@5*, quartile filter and Sim(att) reach the highest precision with 60%.

In summary, the clustering filter retrieves a suitable number of cases for adaptation phase in the CBR. Thus, the final users of CBR have a reduced number of similar cases compared with quartile filter. The clustering filter retrieves in average 6/36 cases for classification tasks and 10/20 cases for regression tasks, while the quartile filter considers the majority of the cases, for example, in classification tasks quartile filter retrieves 30/36, while in regression tasks 19/20 cases.
For similarity mechanisms, the precisions are equals. However, Sim(att) achieves best–ranking quality where the queries are new cases (Queries 3 for classification and regression tasks).

5.6 Summary

This chapter presents the CBR to recommend data cleaning algorithms in classification and regression tasks. The CBR for data cleaning is composed of next phases:

- **Case-base construction:** a case is composed by space of problem and solution. We represented the problem space by the meta-features of the dataset, its attributes, and the target variable. The solution space contains the algorithms of data cleaning used for each dataset. We represent the cases through a Data cleaning ontology (Chapter 4).
- **Case retrieval:** The case retrieval mechanism is composed of a filter and similarity phases. In the first phase, we defined two filter approaches based on clustering and quartile analysis. These filters retrieve a reduced number of relevant cases. The second phase computes a ranking of the retrieved cases by filter approaches, and it scores a similarity between a new case and the retrieved cases.
- **Case reuse:** we proposed a step-by-step to the reuse of a case. If the problem space of new case like to retrieved case, then the old data cleaning solution is copied. In case of problem space of new case is different to the retrieved case, the Data cleaning ontology (Chapter 4) recommends similar data cleaning algorithms to the algorithm proposed in the solution space of retrieved case.
- **Case retain:** to retain a case, we proposed to verify the case quality through human experts supported in three data quality dimensions: Accuracy, Completeness, and Validity.
- **Results:** as mentioned in Subsection 2.1.4, a CBR is essentially centered on retrieval mechanism of cases [48]. Thus, we evaluated the retrieval mechanism through a set of judges. The panel of judges scores the similarity between a query case against all cases of the case-base (ground truth). The results of the retrieval mechanism reach an average precision on judges ranking of 94.5% in top 3 (*P@3*), for top 7 (*P@7*) 84.55%, while in top 10 (*P@10*) 78.35%.

6. Conclusions and future works

In this Chapter, we present the conclusions and propose future works. These are aligned with contributions of this PhD thesis.

6.1 Conclusions

To guarantee a successful knowledge discovery process there are popular data mining methodologies as CRISP-DM and SEMMA. Several knowledge discovery tools are based in these data mining methodologies. According to Gartner 2018 Magic Quadrant for Data Science and Machine Learning Platforms, KNIME [15], RapidMiner [16], SAS [17], Alteryx [18] and H20.ai [19] are the leader tools for knowledge discovery. However, these knowledge discovery tools either do not offer a user oriented process to address data quality issues and mechanisms for the recommendation of the suitable data cleaning algorithms. This fact calls the attention of the authors [104, 105, 106, 107] which they mentioned a list of relevant decisions that must be considered during a knowledge discovery process:

- How to effectively perform data quality verification?
- How to efficiently perform the data preparation phase (i.e. missing values, outliers, duplicate records)?
- Which data cleaning algorithm is most appropriate?
- How to deal with a potential class imbalance problem?
- How to improve the accuracy rate (i.e. error rate)?

To tackle the mentioned challenges, we proposed (i) a conceptual framework user-oriented to address data quality issues, (ii) a case-based reasoning system (CBR) for the recommendation of the suitable data cleaning algorithms and (iii) Data cleaning ontology that gathers the knowledge of the data cleaning algorithms to solve the data quality issues. Thus, we can concluded:

The conceptual framework proposed in Chapter 3 is a useful data cleaning process for classification and regression tasks. We validated the conceptual framework with datasets of the UCI Repository of Machine Learning Databases [158]. We cleaned the datasets following the conceptual framework and applying the data cleaning algorithms. We applied several times these algorithms until obtaining results upper or similar to the obtained by UCI datasets. The cleaned datasets by our conceptual framework were used to train the same algorithms proposed by authors of UCI datasets. In this sense, 85.71% of the classification models achieve the highest precisions and AUC than models proposed by datasets authors, while 90% of the regression models reach Mean Absolute Error less than models proposed by datasets authors. In summary, 87.85% of the models (classification and regression) generated by the datasets cleaned of the conceptual framework (without knowledge of dataset domain) reached good performance compared with the models proposed by datasets authors.

However, the validation process of the CF is not enough due the dataset authors omit details about the process of data preparation as the creation and modification of attributes from original ones, model validation technique (cross-validation, test set, etc.), or experimental configuration of the models. In addition, the original dataset and the dataset cleaned by CF are different. Thus, we proposed mini-challenges (Subsections 3.5.1.3 and 3.5.2.3 with the aim to enrich the validation process. In this way, CF achieved the highest *Accuracy* and *AUC* in 4/6 classification mini-challenges, while regression tasks CF reached the lowest *Mean Absolute Error* in 2/3 mini-challenges. As conclusion, the conceptual framework takes on particular importance when the user has not knowledge about dataset domain. Compared with effort in data preparation and previous domain knowledge by dataset authors, the conceptual framework offers a general data cleaning solution tested on 56 datasets of the UCI Repository.

However, we must know the data cleaning algorithms to apply the suitable method. To solve this problem, we proposed a case-based reasoning (CBR) system (Chapter 5) to recommend the suitable data cleaning algorithms to the inexperienced users of the conceptual framework. As the retrieval is the main phase in a CBR, we focus on the validation of the case retrieval mechanism.This was evaluated through a set of judges from three queries for each knowledge discovery tasks (classification and regression). The first query (Q1) corresponds to a case contained in the case-base, whereas the second query (Q2) is a modified case of the case-base, and the third query (Q3) is a new case. The results of the retrieval mechanism for classification tasks and all queries reach an position precision on judges ranking of 100% in top 1 (P-P@1) and top 2 (P-P@2), while in top 3 (P-P@3) 50%. In case of regression tasks, the retrieval mechanism achieves an position precision of 100% in top 1 (P-P@1), top 2 (P-P@2) and top 3 (P-P@3). In other words, we can guarantee the retrieval of the two most similar cases respect to all queries.

With the aim to support the CBR, we proposed a Data cleaning ontology (DCO). The knowledge acquired in the construction and application of the conceptual framework (data quality issues found in datasets, data cleaning tasks, approaches, and algorithms used) was conceptualized in the Data cleaning ontology for case representation. This reduces considerably the knowledge acquisition bottleneck of data quality in knowledge discovery tasks [251]. Also, the representation of cases through Data cleaning ontology allows the integration between ontologies of specific domains [218] to support some data quality issues, as the selection of relevant attributes based on expert knowledge.
In cancer domain, the ontology developed by [75] is used for selection of relevant attributes and avoid the use of algorithms with high computational complexity in dimensionality reduction tasks. Additionally, Data cleaning ontology supports the case reuse phase of the CBR. DCO recommends similar data cleaning algorithms to the algorithms proposed in the solution of the retrieved case. This allows to the user apply alternative data cleaning algorithms when the recommended data cleaning algorithm obtains poor results.

Finally, our proposal can be improved through domain knowledge. For example, in the dimensionality reduction task, the domain knowledge could support the construction of new attributes based on the original attributes. The new attributes can be relevant to build a model. In case of outliers detection task, the domain knowledge allows to define the values range allowed for each attribute.

6.2 Future works

We propose as future works:

- Increase the number of cases of the case–base. This work is intricate; however, as a first approximation, we suggest to include datasets with unpublished results (in this PhD thesis we only used dataset published in conferences and journals). Thus, the solution spaces of the new cases must guarantee high performance in the evaluation metrics (accuracy, precision, recall, mean absolute errors, etc).

- Add other popular knowledge discovery tasks as clustering to the Conceptual Framework and CBR. This implies to identify new data quality issues, data cleaning tasks, create new case–bases, define new meta–features and update the Data cleaning ontology for the new knowledge discovery tasks.

- For the retain phase, before to save the case into the case– base, we propose to build a formal process for assessment of the quality of adapted cases through methodologies as [21] and several data quality dimensions [7]. The main advantage of using these approaches is the flexibility for identifying cases with poor quality through a set of phases. Additionally, we must consider a set of experts to assess the formal process proposed.

- For the filter approach in the retrieval phase, we propose to use incremental learning to update automatically the cluster and classification models [252, 253]. Thus, the models are updated in an incremental fashion to accommodate new cases without compromising models performance [254].

- Include planners to the Conceptual Framework. These build partial and dynamic solutions based on a set of dataset meta–features and the knowledge of data cleaning tasks represented by the Data cleaning ontology [20].

Bibliography

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC VIEW Sponsored by EMC Corporation*, pp. 1–16, 2012.

[2] H. Hu, Y. Wen, T.-S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *Access, IEEE*, vol. 2, pp. 652–687, 2014.

[3] D. C. Corrales, J. C. Corrales, and A. Ledezma, "How to address the data quality issues in regression models: A guided process for data cleaning," *Symmetry*, vol. 10, no. 4, 2018.

[4] B. Marr, "Big data: 20 mind-boggling facts everyone must read," September 2015. [Online; posted 30-September-2015].

[5] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. New York, N.Y. ; Cambridge: Cambridge University Press, Dec. 2011.

[6] F. Pacheco, C. Rangel, J. Aguilar, M. Cerrada, and J. Altamiranda, "Methodological framework for data processing based on the Data Science paradigm," in *Computing Conference (CLEI), 2014 XL Latin American*, pp. 1–12, Sept. 2014.

[7] L. Sebastian-Coleman, *Measuring data quality for ongoing improvement: a data quality assessment framework*. Newnes, 2012.

[8] N. A. H. Alkharboush, *A data mining approach to improve the automated quality of data*. PhD thesis, Queensland University of Technology, 2014.

[9] Y. Li and K. D. Joshi, "Data cleansing decisions: Insights from discrete-event simulations of firm resources and data quality," *Journal of Organizational Computing and Electronic Commerce*, vol. 22, no. 4, pp. 361–393, 2012.

[10] W. Eckerson, "Data warehousing special report: Data quality and the bottom line," *Applications Development Trends*, vol. 1, 2002.

[11] D. C. Corrales, A. Ledezma, and J. C. Corrales, "A Conceptual Framework for Data Quality in Knowledge Discovery Tasks (FDQ-KDT): A Proposal," *Journal of Computers*, vol. 10, pp. 396–405, Nov. 2015.

[12] T. Yu, N. Chawla, and S. Simoff, *Computational intelligent data analysis for sustainable development*. Chapman and Hall/CRC, 2013.

[13] P. Chapman, *CRISP-DM 1.0: Step-by-step Data Mining Guide*. SPSS, 2000.

[14] "2018 gartner magic quadrant." `https://www.sisense.com/gartner-magic-quadrant-business-intelligence/`. Accessed: 2018-05-31.

[15] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, C. Sieb, K. Thiel, and B. Wiswedel, "Knime: The konstanz information miner," in *Data Analysis, Machine Learning and Applications* (C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, eds.), (Berlin, Heidelberg), pp. 319–326, Springer Berlin Heidelberg, 2008.

[16] M. Hofmann and R. Klinkenberg, *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013.

[17] G. Fernandez, *Data mining using SAS applications*. CRC press, 2010.

[18] R. Baruti, *Learning Alteryx: A Beginner's Guide to Using Alteryx for Self-Service Analytics and Business Intelligence*. Packt Publishing, Limited, 2017.

[19] D. Cook, *Practical Machine Learning with H2O: Powerful, Scalable Techniques for Deep Learning and AI*. O'Reilly Media, Incorporated, 2016.

[20] F. Serban, J. Vanschoren, J.-U. Kietz, and A. Bernstein, "A survey of intelligent assistants for data analysis," *ACM Comput. Surv.*, vol. 45, pp. 31:1–31:35, July 2013.

[21] Y. Jabareen, "Building a conceptual framework: philosophy, definitions, and procedure," *International Journal of Qualitative Methods*, vol. 8, no. 4, pp. 49–62, 2009.

[22] U. Fayyad, G. Piatetsky-shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, pp. 37–54, 1996.

[23] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[24] C. O'Neil and R. Schutt, *Doing Data Science: Straight Talk from the Frontline*. O'Reilly Media, 1 edition ed., Nov. 2013.

[25] V. Dhar, "Data science and prediction," *Commun. ACM*, vol. 56, pp. 64–73, Dec. 2013.

[26] I. Caballero, E. Verbo, C. Calero, and M. Piattini, "A data quality measurement information model based on iso/iec 15939.," in *ICIQ*, pp. 393–408, Cambridge, MA, 2007.

[27] D. P. Ballou and H. L. Pazer, "Modeling data and process quality in multi-input, multi-output information systems," *Manage. Sci.*, vol. 31, pp. 150–162, Feb. 1985.

[28] L. Berti-Équille, *Measuring and Modelling Data Quality for Quality-Awareness in Data Mining*, pp. 101–126. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[29] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Commun. ACM*, vol. 45, pp. 211–218, Apr. 2002.

[30] K. Kerr and T. Norris, "The development of a healthcare data quality framework and strategy.," in *Proceedings of the Ninth International Conference on Information Quality (ICIQ-04)*, pp. 218–233, 2004.

[31] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *Journal of management information systems*, vol. 12, no. 4, pp. 5–33, 1996.

[32] M. J. Eppler and D. Wittig, "Conceptualizing information quality: A review of information quality frameworks from the last ten years.," in *IQ*, pp. 83–96, 2000.

[33] Y. Toninato and A. Joppe, "Agile data quality framework," *https://www2.deloitte.com/be/en/pages/technology/articles/agile-data-quality.html*, 2018 (accessed July 15, 2018).

[34] N. Micic, D. Neagu, F. Campean, and E. H. Zadeh, "Towards a data quality framework for heterogeneous data," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and*

Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 155–162, June 2017.

[35] B. H. Guo and Y. M. Goh, "Ontology for design of active fall protection systems," *Automation in Construction*, pp. –, 2017.

[36] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?," *International Journal of Human-Computer Studies*, vol. 43, no. 5, pp. 907 – 928, 1995.

[37] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review*, vol. 11, no. 02, pp. 93–136, 1996.

[38] S. Mario, M. Dorian, and A. M. Myrup, "Relationships between the concepts in the design ontology," *Guidelines for a Decision Support Method Adapted to NPD Processes*, 2007.

[39] A. C. Varzi, "Spatial reasoning and ontology: parts, wholes, and locations," in *Handbook of spatial logics*, pp. 945–1038, Springer, 2007.

[40] J. F. Sowa, "Conceptual structures: information processing in mind and machine," 1983.

[41] A. Borgida, T. J. Walsh, H. Hirsh, *et al.*, "Towards measuring similarity in description logics.," *Description Logics*, vol. 147, 2005.

[42] R. M. Smullyan, *First-order logic*. Courier Corporation, 1995.

[43] S. Bechhofer, "Owl: Web ontology language," in *Encyclopedia of database systems*, pp. 2008–2009, Springer, 2009.

[44] D. McGuinness and F. van Harmelen, "OWL web ontology language overview," W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[45] C. Roussey, F. Pinet, M. A. Kang, and O. Corcho, *An Introduction to Ontologies and Ontology Engineering*, pp. 9–38. London: Springer London, 2011.

[46] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Commun.*, vol. 7, pp. 39–59, Mar. 1994.

[47] D. B. Leake, *Case-Based Reasoning: Experiences, Lessons and Future Directions*. Cambridge, MA, USA: MIT Press, 1st ed., 1996.

[48] M. M. Richter and R. O. Weber, *Case-Based Reasoning: A Textbook*. Springer Publishing Company, Incorporated, 2013.

[49] J. McCarthy, "The frame problem today," in *The Frame Problem in Artificial Intelligence* (F. M. BROWN, ed.), p. 3, Morgan Kaufmann, 1987.

[50] J. A. Recio-García, P. A. González-Calero, and B. Díaz-Agudo, "jcolibri2: A framework for building case-based reasoning systems," *Science of Computer Programming*, vol. 79, pp. 126 – 145, 2014. Experimental Software and Toolkits (EST 4): A special issue of the Workshop on Academic Software Development Tools and Techniques (WASDeTT-3 2010).

[51] A. Wyner, R. Mochales-Palau, M.-F. Moens, and D. Milward, "Approaches to text mining arguments from legal cases," in *Semantic processing of legal texts*, pp. 60–79, Springer, 2010.

[52] B. Díaz-Agudo and P. A. González-Calero, "An architecture for knowledge intensive cbr systems," in *European workshop on advances in case-based reasoning*, pp. 37–48, Springer, 2000.

[53] A. Fornells, E. Golobardes, D. Vernet, and G. Corral, "Unsupervised case memory organization: Analysing computational time and soft computing capabilities," in *Proceedings of the 8th European Conference on Advances in Case-Based Reasoning*, ECCBR'06, (Berlin, Heidelberg), pp. 241–255, Springer-Verlag, 2006.

[54] E. Plaza and L. McGinty, "Distributed case-based reasoning," *The Knowledge engineering review*, vol. 20, no. 3, pp. 261–265, 2005.

[55] F. Chiang and S. Sitaramachandran, *A Data Quality Framework for Customer Relationship Analytics*, pp. 366–378. Cham: Springer International Publishing, 2015.

[56] H. Galhard, D. Florescu, D. Shasha, and E. Simon, "An extensible framework for data cleaning," in *Data Engineering, 2000. Proceedings. 16th International Conference on*, pp. 312–312, 2000.

[57] S. de F. Mendes Sampaio, C. Dong, and P. Sampaio, "Dq2s ? a framework for data quality-aware information management," *Expert Systems with Applications*, vol. 42, no. 21, pp. 8304 – 8326, 2015.

[58] W. Li and L. Lei, *An Object-Oriented Framework for Data Quality Management of Enterprise Data Warehouse*, pp. 1125–1129. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[59] I. Taleb, R. Dssouli, and M. A. Serhani, "Big data pre-processing: A quality framework," in *2015 IEEE International Congress on Big Data*, pp. 191–198, June 2015.

[60] A. P. Reimer, A. Milinovich, and E. A. Madigan, "Data quality assessment framework to assess electronic medical record data for use in research," *International Journal of Medical Informatics*, vol. 90, pp. 40 – 47, 2016.

[61] O. Almutiry, G. Wills, and A. Alwabel, "Toward a framework for data quality in cloud-based health information system," in *Information Society (i-Society), 2013 International Conference on*, pp. 153–157, June 2013.

[62] D. G. Arts, N. F. De Keizer, and G.-J. Scheffer, "Defining and improving data quality in medical registries: a literature review, case study, and generic framework," *Journal of the American Medical Informatics Association*, vol. 9, no. 6, pp. 600–611, 2002.

[63] P. Myrseth, J. Stang, and V. Dalberg, "A data quality framework applied to e-government metadata: A prerequsite to establish governance of interoperable e-services," in *E -Business and E -Government (ICEE), 2011 International Conference on*, pp. 1–4, May 2011.

[64] A. Vetro, L. Canova, M. Torchiano, C. O. Minotas, R. Iemma, and F. Morando, "Open data quality measurement framework: Definition and application to open government data," *Government Information Quarterly*, vol. 33, no. 2, pp. 325 – 337, 2016.

[65] P. H. S. Panahy, F. Sidi, L. S. Affendey, M. A. Jabar, H. Ibrahim, and A. Mustapha, "A framework to construct data quality dimensions relationships," *Indian Journal of Science and Technology*, vol. 6, no. 5, 2013.

[66] R. Y. Wang, V. C. Storey, and C. P. Firth, "A framework for analysis of data quality research," *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, pp. 623–640, Aug 1995.

[67] K. Rasta, T. H. Nguyen, and A. Prinz, "A framework for data quality handling in enterprise service bus," in *Innovative Computing Technology (INTECH), 2013 Third International Conference on*, pp. 491–497, Aug 2013.

[68] X. Wang, H. J. Hamilton, and Y. Bither, "An ontology-based approach to data cleaning," Tech. Rep. CS-2005-05, Department of Computer Science, University of Regina, 2005.

[69] R. Almeida, P. Oliveira, L. Braga, and J. Barroso, "Ontologies for reusing data cleaning knowledge," in *2012 IEEE Sixth International Conference on Semantic Computing*, pp. 238–241, Sept 2012.

[70] S. Brüggemann, *Rule Mining for Automatic Ontology Based Data Cleaning*, pp. 522–527. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[71] Z. Kedad and E. Métais, *Ontology-Based Data Cleaning*, pp. 137–149. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[72] S. G. Johnson, S. Speedie, G. Simon, V. Kumar, and B. L. Westra, "A Data Quality Ontology for the Secondary Use of EHR Data," *AMIA Annual Symposium Proceedings*, vol. 2015, pp. 1937–1946, Nov. 2015.

[73] R. G. Abarza, R. Motz, and A. Urrutia, "Quality assessment using data ontologies," in *2014 33rd International Conference of the Chilean Computer Science Society (SCCC)*, pp. 30–33, Nov 2014.

[74] L. F. Garcia, V. M. Graciolli, L. F. D. Ros, and M. Abel, "An ontology-based conceptual framework to improve rock data quality in reservoir models," in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1084–1088, Nov 2016.

[75] A. da Silva Jacinto, R. da Silva Santos, and J. M. P. de Oliveira, "Automatic and semantic pre-selection of features using ontology for data mining on data sets related to cancer," in *International Conference on Information Society (i-Society 2014)*, pp. 282–287, Nov 2014.

[76] A. Coulet, M. Smaïl-Tabbone, P. Benlian, A. Napoli, and M.-D. Devignes, "Ontology-guided data preparation for discovering genotype-phenotype relationships," *BMC Bioinformatics*, vol. 9, p. S3, Apr. 2008.

[77] C. Diamantini, D. Potena, and E. Storti, "Kddonto: An ontology for discovery and composition of kdd algorithms," in *Proceedings of the ECML/PKDD09 Workshop on 3rd generation Data Mining (SoKD-09)*, pp. 13–24, 2009.

[78] P. Panov, S. Džeroski, and L. Soldatova, "Ontodm: An ontology of data mining," in *2008 IEEE International Conference on Data Mining Workshops*, pp. 752–760, Dec 2008.

[79] C. M. Keet, A. ?awrynowicz, C. d?Amato, A. Kalousis, P. Nguyen, R. Palma, R. Stevens, and M. Hilario, "The data mining {OPtimization} ontology," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 32, pp. 43 – 53, 2015.

[80] M. Hilario, A. Kalousis, P. Nguyen, and A. Woznica, "A data mining ontology for algorithm selection and meta-mining," in *Proceedings of the ECML/PKDD09 Workshop on 3rd generation Data Mining (SoKD-09)*, pp. 76–87, 2009.

[81] S. Nadal, O. Romero, A. Abelló, P. Vassiliadis, and S. Vansummeren, "An integration-oriented ontology to govern evolution in big data ecosystems," *Information Systems*, 2018.

[82] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web. (Advanced Information and Knowledge Processing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[83] A. Bandrowski, R. Brinkman, M. Brochhausen, M. H. Brush, B. Bug, M. C. Chibucos, K. Clancy, M. Courtot, D. Derom, M. Dumontier, *et al.*, "The ontology for biomedical investigations," *PloS one*, vol. 11, no. 4, p. e0154556, 2016.

[84] L. N. Soldatova and R. D. King, "An ontology of scientific experiments," *Journal of the Royal Society Interface*, vol. 3, no. 11, pp. 795–803, 2006.

[85] H. Cai, X. Zhang, Y. Zhang, Z. Wang, and B. Hu, "A case-based reasoning model for depression based on three-electrode eeg data," *IEEE Transactions on Affective Computing*, pp. 1–1, 2018.

[86] D. Brown, A. Aldea, R. Harrison, C. Martin, and I. Bayley, "Temporal case-based reasoning for type 1 diabetes mellitus bolus insulin decision support," *Artificial Intelligence in Medicine*, vol. 85, pp. 28 – 42, 2018.

[87] R. Saraiva, M. Perkusich, L. Silva, H. Almeida, C. Siebra, and A. Perkusich, "Early diagnosis of gastrointestinal cancer by using case-based and rule-based reasoning," *Expert Systems with Applications*, vol. 61, pp. 192 – 202, 2016.

[88] D. Gu, C. Liang, and H. Zhao, "A case-based reasoning system based on weighted heterogeneous value distance metric for breast cancer diagnosis," *Artificial Intelligence in Medicine*, vol. 77, pp. 31 – 47, 2017.

[89] A. Yan, H. Yu, and D. Wang, "Case-based reasoning classifier based on learning pseudo metric retrieval," *Expert Systems with Applications*, vol. 89, pp. 91 – 98, 2017.

[90] A. Yan, K. Zhang, Y. Yu, and P. Wang, "An attribute difference revision method in case-based reasoning and its application," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 212 – 219, 2017.

[91] H. Zhao, J. Liu, W. Dong, X. Sun, and Y. Ji, "An improved case-based reasoning method and its application on fault diagnosis of tennessee eastman process," *Neurocomputing*, vol. 249, pp. 266 – 276, 2017.

[92] F. Sartori, A. Mazzucchelli, and A. D. Gregorio, "Bankruptcy forecasting using case-based reasoning: The creperie approach," *Expert Systems with Applications*, vol. 64, pp. 400 – 411, 2016.

[93] M. Relich and P. Pawlewski, "A case-based reasoning approach to cost estimation of new product development," *Neurocomputing*, vol. 272, pp. 40 – 45, 2018.

[94] A. Udoh and O. Daramola, "A semantic case-based reasoning framework for enterprise team selection," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, pp. 1–6, Oct 2017.

[95] S. S. Mustapha, "Case-based reasoning for identifying knowledge leader within online community," *Expert Systems with Applications*, vol. 97, pp. 244 – 252, 2018.

[96] A. D. Renzis, M. Garriga, A. Flores, A. Cechich, and A. Zunino, "Case-based reasoning for web service discovery and selection," *Electronic Notes in Theoretical Computer Science*, vol. 321, pp. 89 – 112, 2016. CLEI 2015, the XLI Latin American Computing Conference.

[97] H. Y. Abutair and A. Belghith, "Using case-based reasoning for phishing detection," *Procedia Computer Science*, vol. 109, pp. 281 – 288, 2017. 8th International Conference on Ambient Systems, Networks and Technologies, ANT-2017 and the 7th International Conference on Sustainable Energy Information Technology, SEIT 2017, 16-19 May 2017, Madeira, Portugal.

[98] J. Ahn, M. Park, H.-S. Lee, S. J. Ahn, S.-H. Ji, K. Song, and B.-S. Son, "Covariance effect analysis of similarity measurement methods for early construction cost estimation using case-based reasoning," *Automation in Construction*, vol. 81, pp. 254 – 266, 2017.

[99] E. Lupiani, J. M. Juarez, J. Palma, and R. Marin, “Monitoring elderly people at home with temporal case-based reasoning,” *Knowledge-Based Systems*, vol. 134, pp. 116 – 134, 2017.

[100] R. Engels, “Planning tasks for knowledge discovery in databases; performing task-oriented user-guidance,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD’96, pp. 170–175, AAAI Press, 1996.

[101] R. Wirth, C. Shearer, U. Grimmer, T. Reinartz, J. Schlösser, C. Breitner, R. Engels, and G. Lindner, *Towards process-oriented tool support for knowledge discovery in databases*, pp. 243–253. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997.

[102] G. Lindner and R. Studer, *AST: Support for Algorithm Selection with a CBR Approach*, pp. 418–423. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[103] K. Morik and M. Scholz, *The MiningMart Approach to Knowledge Discovery in Databases*, pp. 47–65. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[104] M. Charest, S. Delisle, and O. Cervantes, “Design considerations for a cbr-based intelligent data mining assistant,” 2006.

[105] M. Charest, S. Delisle, O. Cervantes, and Y. Shen, “Invited paper: Intelligent data mining assistance via cbr and ontologies,” in *17th International Workshop on Database and Expert Systems Applications (DEXA’06)*, pp. 593–597, 2006.

[106] M. Charest and S. Delisle, “Ontology-guided intelligent data mining assistance: Combining declarative and procedural knowledge.,” in *Artificial Intelligence and Soft Computing*, pp. 9–14, 2006.

[107] M. Charest, S. Delisle, O. Cervantes, and Y. Shen, “Bridging the gap between data mining and decision support: A case-based reasoning and ontology approach,” *Intell. Data Anal.*, vol. 12, pp. 211–236, Apr. 2008.

[108] M. Choinski and J. A. Chudziak, “Ontological learning assistant for knowledge discovery and data mining,” in *2009 International Multiconference on Computer Science and Information Technology*, pp. 147–155, Oct 2009.

[109] K. Gibert, M. Sànchez-Marrè, and V. Codina, “Choosing the right data mining technique: classification of methods and intelligent recommendation,” in *International Congress on Environmental Modelling and Software*, 2010.

[110] S. El-Sappagh, M. Elmogy, A. Riad, H. Zaghlol, and F. A. Badria, "Ehr data preparation for case based reasoning construction," in *International Conference on Advanced Machine Learning Technologies and Applications*, pp. 483–497, Springer, 2014.

[111] L. Giustarini, O. Parisot, M. Ghoniem, R. Hostache, I. Trebs, and B. Otjacques, "A user-driven case-based reasoning tool for infilling missing values in daily mean river flow records," *Environmental Modelling and Software*, vol. 82, pp. 308 – 320, 2016.

[112] H. Zhang and G. Dai, "Research on traffic decision making method based on image analysis case based reasoning," *Optik*, vol. 158, pp. 908 – 914, 2018.

[113] H. Zhao, H. Chen, W. Dong, X. Sun, and Y. Ji, "Fault diagnosis of rail turnout system based on case-based reasoning with compound distance methods," in *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 4205–4210, May 2017.

[114] F. Tempola, A. Arief, and M. Muhammad, "Combination of case-based reasoning and nearest neighbour for recommendation of volcano status," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pp. 348–352, Nov 2017.

[115] D. L. Rubin, H. Knublauch, R. W. Fergerson, O. Dameron, and M. A. Musen, "Protege-owl: Creating ontology-driven reasoning applications with the web ontology language," in *AMIA Annual Symposium Proceedings*, vol. 2005, p. 1179, American Medical Informatics Association, 2005.

[116] G. Piateski and W. Frawley, *Knowledge Discovery in Databases*. Cambridge, MA, USA: MIT Press, 1991.

[117] R. Schutt and C. O'Neil, *Doing Data Science: Straight Talk from the Frontline*. O'Reilly Media, Inc., 2013.

[118] D. C. y Agapito Ledezma y Juan Corrales, "A systematic review of data quality issues in knowledge discovery tasks," *Revista Ingenierias Universidad de Medellín*, vol. 15, no. 28, 2016.

[119] M. Bosu and S. Macdonell, "A Taxonomy of Data Quality Challenges in Empirical Software Engineering," in *Software Engineering Conference (ASWEC), 2013 22nd Australian*, pp. 97–106, June 2013.

[120] M. F. Bosu and S. G. MacDonell, "A taxonomy of data quality challenges in empirical software engineering," in *2013 22nd Australian Software Engineering Conference*, pp. 97–106, June 2013.

[121] H. Xiong, G. Pandey, M. Steinbach, and V. Kumar, "Enhancing data analysis with noise removal," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 304–319, March 2006.

[122] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, July 2009.

[123] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Information Sciences*, vol. 233, pp. 25 – 35, 2013.

[124] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.

[125] V. Barnett, T. Lewis, *et al.*, *Outliers in statistical data*, vol. 3. Wiley New York, 1994.

[126] R. A. Johnson, D. W. Wichern, *et al.*, *Applied multivariate statistical analysis*, vol. 4. Prentice-Hall New Jersey, 2014.

[127] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Science and Information Conference (SAI), 2014*, pp. 372–378, Aug 2014.

[128] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data Classification: Algorithms and Applications*, p. 37, 2014.

[129] L.Ladha and T.Deepa, "Feature selection methods and algorithms," *International Journal on Computer Science and Engineering*, vol. 3, no. 5, pp. 1787?–1797, 2011.

[130] A. Hasan and M. A. Adnan, "High dimensional microarray data classification using correlation based feature selection," in *2012 International Conference on Biomedical Engineering (ICoBE)*, pp. 319–321, Feb 2012.

[131] A. P. Kumar and P. Valsala, "Feature selection for high dimensional dna microarray data using hybrid approaches," *Bioinformation*, vol. 9, no. 16, p. 824, 2013.

[132] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263–1284, Sept 2009.

[133] I. Chaïri, S. Alaoui, and A. Lyhyaoui, "Learning from imbalanced data using methods of sample selection," in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, pp. 254–257, May 2012.

[134] D. C. Corrales, A. Figueroa, A. Ledezma, and J. C. Corrales, "An empirical multi-classifier for coffee rust detection in colombian crops," in *International Conference on Computational Science and Its Applications*, pp. 60–74, Springer, 2015.

[135] D. C. Corrales, A. Ledezma, A. J. Peña, J. Hoyos, A. Figueroa, and J. C. Corrales, "A new dataset for coffee rust detection in colombian crops base on classifiers," *Sistemas & Telemática*, vol. 12, no. 29, pp. 9–23, 2014.

[136] D. C. Corrales, G. Gutierrez, J. P. Rodriguez, A. Ledezma, and J. C. Corrales, "Lack of data: Is it enough estimating the coffee rust with meteorological time series?," in *Computational Science and Its Applications – ICCSA 2017* (O. Gervasi, B. Murgante, S. Misra, G. Borruso, C. M. Torre, A. M. A. Rocha, D. Taniar, B. O. Apduhan, E. Stankova, and A. Cuzzocrea, eds.), (Cham), pp. 3–16, Springer International Publishing, 2017.

[137] F. Hakimpour and A. Geppert, "Resolving semantic heterogeneity in schema integration," in *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, (New York, NY, USA), pp. 297–308, ACM, 2001.

[138] M. Finger and F. S. D. Silva, "Temporal data obsolescence: modelling problems," in *Proceedings. Fifth International Workshop on Temporal Representation and Reasoning (Cat. No.98EX157)*, pp. 45–50, May 1998.

[139] A. Maydanchik, *Data Quality Assessment*. Data quality for practitioners series, Technics Publications, 2007.

[140] T. Aljuaid and S. Sasi, "Proper imputation techniques for missing values in data sets," in *2016 International Conference on Data Science and Engineering (ICDSE)*, pp. 1–5, Aug 2016.

[141] K. Strike, K. E. Emam, and N. Madhavji, "Software cost estimation with incomplete data," *IEEE Transactions on Software Engineering*, vol. 27, pp. 890–908, Oct 2001.

[142] J. W. Grzymala-Busse and M. Hu, "A comparison of several approaches to missing attribute values in data mining," in *Rough Sets and Current Trends in Computing* (W. Ziarko and Y. Yao, eds.), (Berlin, Heidelberg), pp. 378–385, Springer Berlin Heidelberg, 2001.

[143] M. Magnani, "Techniques for dealing with missing data in knowledge discovery tasks," *Obtido http://magnanim. web. cs. unibo. it/index. html*, vol. 15, no. 01, p. 2007, 2004.

[144] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, pp. 93–104, ACM, 2000.

[145] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *KDD-96 Proceedings*, 1996.

[146] H.-P. Kriegel, A. Zimek, *et al.*, "Angle-based outlier detection in high-dimensional data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 444–452, ACM, 2008.

[147] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Advances in knowledge discovery and data mining," ch. From Data Mining to Knowledge Discovery: An Overview, pp. 1–34, Menlo Park, CA, USA: American Association for Artificial Intelligence, 1996.

[148] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16 – 28, 2014. 40th-year commemorative issue.

[149] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1, pp. 245 – 271, 1997.

[150] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[151] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[152] H. He and Y. Ma, *Imbalanced learning: foundations, algorithms, and applications*. John Wiley and Sons, 2013.

[153] D. Devi, B. Purkayastha, *et al.*, "Redundancy-driven modified tomek-link based undersampling: A solution to class imbalance," *Pattern Recognition Letters*, vol. 93, pp. 3–12, 2017.

[154] B. Frenay and M. Verleysen, “Classification in the presence of label noise: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 845–869, May 2014.

[155] L. Huang, H. Jin, P. Yuan, and F. Chu, “Duplicate records cleansing with length filtering and dynamic weighting,” in *2008 Fourth International Conference on Semantics, Knowledge and Grid*, pp. 95–102, Dec 2008.

[156] N. Verbiest, E. Ramentol, C. Cornelis, and F. Herrera, *Improving SMOTE with Fuzzy Rough Prototype Selection to Detect Noise in Imbalanced Classification Data*, pp. 169–178. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[157] A. P. Jacquemin and C. H. Berry, “Entropy measure of diversification and corporate growth,” *The journal of industrial economics*, pp. 359–369, 1979.

[158] A. Asuncion and D. Newman, “Uci machine learning repository. irvine, ca: University of california, school of information and computer science,” *URL [http://www. ics. uci. edu/˜ mlearn/MLRepository. html]*, 2007.

[159] A. Reiss and D. Stricker, “Creating and benchmarking a new dataset for physical activity monitoring,” in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA ’12, (New York, NY, USA), pp. 40:1–40:8, ACM, 2012.

[160] K. Singh, R. Kaur, and D. Kumar, “Comment volume prediction using neural networks and decision trees,” in *Proceedings of the 2015 17th UKSIM-AMSS International Conference on Modelling and Simulation*, UKSIM ’15, (Washington, DC, USA), pp. 15–20, IEEE Computer Society, 2015.

[161] F. E. Grubbs, “Procedures for detecting outlying observations in samples,” *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[162] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, “Tackling the poor assumptions of naive bayes text classifiers,” in *In Proceedings of the Twentieth International Conference on Machine Learning*, pp. 616–623, 2003.

[163] J. G. Colonna, M. Cristo, M. Salvatierra, and E. F. Nakamura, “An incremental technique for real-time bioacoustic signal segmentation,” *Expert Systems with Applications*, vol. 42, no. 21, pp. 7367 – 7374, 2015.

[164] J. G. Colonna, J. Gama, and E. F. Nakamura, “How to correctly evaluate an automatic bioacoustics classification method,” in *Advances in Artificial Intelligence* (O. Luaces, J. A. Gámez, E. Barrenechea, A. Troncoso, M. Galar,

H. Quintián, and E. Corchado, eds.), (Cham), pp. 37–47, Springer International Publishing, 2016.

[165] J. G. Colonna, J. Gama, and E. F. Nakamura, "Recognizing family, genus, and species of anuran using a hierarchical classification approach," in *Discovery Science* (T. Calders, M. Ceci, and D. Malerba, eds.), (Cham), pp. 198–212, Springer International Publishing, 2016.

[166] F. Thabtah, "Autism spectrum disorder screening: Machine learning adaptation and dsm-5 fulfillment," in *Proceedings of the 1st International Conference on Medical and Health Informatics 2017*, ICMHI '17, (New York, NY, USA), pp. 1–6, ACM, 2017.

[167] J. Estrela da Silva, J. P. Marques de Sá, and J. Jossinet, "Classification of breast tissue by electrical impedance spectroscopy," *Medical and Biological Engineering and Computing*, vol. 38, pp. 26–30, Jan 2000.

[168] D. Ayres-de Campos, J. Bernardes, A. Garrido, J. Marques-de Sa, and L. Pereira-Leite, "Sisporto 2.0: a program for automated analysis of cardiotocograms," *Journal of Maternal-Fetal Medicine*, vol. 9, no. 5, pp. 311–318, 2000.

[169] I.-C. Yeh and C. hui Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 2473 – 2480, 2009.

[170] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[171] K. Zhang and W. Fan, "Forecasting skewed biased stochastic ozone days: analyses, solutions and beyond," *Knowledge and Information Systems*, vol. 14, pp. 299–326, Mar 2008.

[172] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5948–5959, 2014.

[173] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and {CO2} measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28 – 39, 2016.

[174] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," *Neural Computing and Applications*, vol. 25, pp. 443–458, Aug 2014.

[175] R. Misir, M. Mitra, and R. Samanta, "A reduced set of features for chronic kidney disease prediction," *Journal of Pathology Informatics*, vol. 8, no. 1, p. 24, 2017.

[176] M. Zikeba, S. K. Tomczak, and J. M. Tomczak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert Systems with Applications*, 2016.

[177] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22 – 31, 2014.

[178] K. Mansouri, T. Ringsted, D. Ballabio, R. Todeschini, and V. Consonni, "Quantitative structure–activity relationship models for ready biodegradability of chemicals," *Journal of chemical information and modeling*, vol. 53, no. 4, pp. 867–878, 2013.

[179] K. Fernandes, J. S. Cardoso, and J. Fernandes, "Transfer learning with partial observability applied to cervical cancer screening," in *Pattern Recognition and Image Analysis* (L. A. Alexandre, J. Salvador Sánchez, and J. M. F. Rodrigues, eds.), (Cham), pp. 243–250, Springer International Publishing, 2017.

[180] H. K. Fatlawi, "Enhanced classification model for cervical cancer dataset based on cost sensitive classifier," *International Journal of Computer Techniques*.

[181] J. Kabiesz, B. Sikora, M. Sikora, and Ł. Wróbel, "Application of rule-based models for seismic hazard prediction in coal mines.," *Acta Montanistica Slovaca*, vol. 18, no. 4, 2013.

[182] A. R. da Rocha Neto and G. de Alencar Barreto, "On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis," *IEEE Latin America Transactions*, vol. 7, pp. 487–496, Aug 2009.

[183] A. R. da Rocha Neto, R. Sousa, G. de A. Barreto, and J. S. Cardoso, "Diagnostic of pathology on the vertebral column with embedded reject option," in *Pattern Recognition and Image Analysis* (J. Vitrià, J. M. Sanches, and

M. Hernández, eds.), (Berlin, Heidelberg), pp. 588–595, Springer Berlin Heidelberg, 2011.

[184] A. Tsanas, M. A. Little, C. Fox, and L. O. Ramig, "Objective automatic assessment of rehabilitative speech treatment in parkinson's disease," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, pp. 181–190, Jan 2014.

[185] ACM, "Annual data mining and knowledge discovery competition (kdd cup)," *http://www.kdd.org/kdd-cup*, 2018 (accessed July 16, 2018).

[186] M. Shahrokh Esfahani and E. R. Dougherty, "Effect of separate sampling on classification accuracy," *Bioinformatics*, vol. 30, no. 2, pp. 242–250, 2013.

[187] A. Ross and V. L. Willson, *Paired Samples T-Test*, pp. 17–19. Rotterdam: SensePublishers, 2017.

[188] R. C. Team *et al.*, "R: A language and environment for statistical computing," 2013.

[189] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1, pp. 278–282, IEEE, 1995.

[190] F. Faubel, J. McDonough, and D. Klakow, "Bounded conditional mean imputation with gaussian mixture models: A reconstruction approach to partly occluded features," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 3869–3872, IEEE, 2009.

[191] E. F. Castillo, W. F. Gonzales, D. C. Corrales, I. D. Lopez, M. G. Hoyos, A. Figueroa, and J. C. Corrales, "Water quality warnings based on cluster analysis in colombian river basins," *Sistemas and Telematica*, vol. 13, no. 33, pp. 9–26, 2015.

[192] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data*, MineNet '06, (New York, NY, USA), pp. 281–286, ACM, 2006.

[193] D. Bitton and D. J. DeWitt, "Duplicate record elimination in large data files," *ACM Trans. Database Syst.*, vol. 8, pp. 255–265, June 1983.

[194] D. C. Corrales, E. Lasso, A. Ledezma, and J. C. Corrales, “Feature selection for classification tasks: Expert knowledge or traditional methods?,” *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–11.

[195] M. Kuhn, “Caret package,” *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008.

[196] R. L. Plackett, “Karl pearson and the chi-squared test,” *International Statistical Review/Revue Internationale de Statistique*, pp. 59–72, 1983.

[197] J. T. Kent, “Information gain and a general measure of correlation,” *Biometrika*, vol. 70, no. 1, pp. 163–173, 1983.

[198] T. M. Mitchell *et al.*, “Machine learning. 1997,” *Burr Ridge, IL: McGraw Hill*, vol. 45, no. 37, pp. 870–877, 1997.

[199] D. Roobaert, G. Karakoulas, and N. V. Chawla, “Information gain, correlation and support vector machines,” pp. 463–470, 2006.

[200] S. Sathyadevan and M. A. Chaitra, “Airfoil self noise prediction using linear regression approach,” in *Computational Intelligence in Data Mining - Volume 2* (L. C. Jain, H. S. Behera, J. K. Mandal, and D. P. Mohapatra, eds.), (New Delhi), pp. 551–561, Springer India, 2015.

[201] X. Liang, T. Zou, B. Guo, S. Li, H. Zhang, S. Zhang, H. Huang, and S. X. Chen, “Assessing beijing’s pm2. 5 pollution: severity, weather impact, apec and winter heating,” *Proc. R. Soc. A*, vol. 471, no. 2182, p. 20150257, 2015.

[202] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari, “Machine learning approaches for improving condition-based maintenance of naval propulsion plants,” *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 230, no. 1, pp. 136–153, 2016.

[203] H. Fanaee-T and J. Gama, “Event labeling combining ensemble detectors and background knowledge,” *Progress in Artificial Intelligence*, vol. 2, pp. 113–127, Jun 2014.

[204] S. D. Vito, G. Fattoruso, M. Pardo, F. Tortorella, and G. D. Francia, “Semi-supervised learning techniques in artificial olfaction: A novel approach to classification problems and drift counteraction,” *IEEE Sensors Journal*, vol. 12, pp. 3215–3224, Nov 2012.

[205] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario," *Sensors and Actuators B: Chemical*, vol. 129, no. 2, pp. 750–757, 2008.

[206] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81 – 97, 2017.

[207] S. Moro, P. Rita, and B. Vala, "Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach," *Journal of Business Research*, vol. 69, no. 9, pp. 3341 – 3351, 2016.

[208] K. Buza, "Feedback prediction for blogs," in *Data Analysis, Machine Learning and Knowledge Discovery* (M. Spiliopoulou, L. Schmidt-Thieme, and R. Janning, eds.), (Cham), pp. 145–152, Springer International Publishing, 2014.

[209] P. Cortez and A. d. J. R. Morais, "A data mining approach to predict forest fires using meteorological data," 2007.

[210] F. Zamora-Martinez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "Online learning of indoor temperature forecasting models towards energy efficiency," *Energy and Buildings*, vol. 83, pp. 162 – 172, 2014. SCIENCE BEHIND AND BEYOND THE SOLAR DECATHLON EUROPE 2012.

[211] M. R. Bautista-Zambrana, "Methodologies to build ontologies for terminological purposes," *Procedia - Social and Behavioral Sciences*, vol. 173, pp. 264 – 269, 2015. 32nd International Conference of the Spanish Association of Applied Linguistics (AESLA): Language Industries and Social Change. 3-5 April 2014, Seville, {SPAIN}.

[212] M. Uschold and M. King, "Towards a methodology for building ontologies," in *In Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, 1995.

[213] S. Staab, R. Studer, H. P. Schnurr, and Y. Sure, "Knowledge processes and ontologies," *IEEE Intelligent Systems*, vol. 16, pp. 26–34, Jan 2001.

[214] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05*, 2001.

[215] N. Aussenac-Gilles, S. Despres, and S. Szulman, “The terminae method and platform for ontology engineering from texts,” in *Proceedings of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, (Amsterdam, The Netherlands, The Netherlands), pp. 199–223, IOS Press, 2008.

[216] R. Temmerman and K. Kerremans, “Termontography: Ontology building and the sociocognitive approach to terminology description,” in *Proceedings of CIL17. Prag: Matfyzpress, MFF UK*, 2017.

[217] I. Horrocks, P. F. Patel-Schneider, H. Bole, S. Tabet, B. Grosof, and M. Dean, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML.” `https://www.w3.org/Submission/SWRL/`, 2004. [Online; accessed 2017-03-01].

[218] S. H. El-Sappagh and M. Elmogy, “Case based reasoning: Case representation methodologies,” *International Journal of Advanced Computer Science & Applications*, vol. 1, no. 6, pp. 192–208, 2015.

[219] M. Zi?ba, S. K. Tomczak, and J. M. Tomczak, “Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction,” *Expert Systems with Applications*, vol. 58, pp. 93 – 101, 2016.

[220] A. Filchenkov and A. Pendryak, “Datasets meta-feature description for recommending feature selection algorithm,” in *2015 Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)*, pp. 11–18, Nov 2015.

[221] G. Wang, Q. Song, H. Sun, X. Zhang, B. Xu, and Y. Zhou, “A feature subset selection algorithm automatic recommendation method,” *J. Artif. Int. Res.*, vol. 47, pp. 1–34, May 2013.

[222] C. Castiello, G. Castellano, and A. M. Fanelli, “Meta-data: Characterization of input features for meta-learning,” in *Modeling Decisions for Artificial Intelligence* (V. Torra, Y. Narukawa, and S. Miyamoto, eds.), (Berlin, Heidelberg), pp. 457–468, Springer Berlin Heidelberg, 2005.

[223] G. Lindner, D. Ag, and R. Studer, “Ast: Support for algorithm selection with a cbr approach,” in *Recent Advances in Meta-Learning and Future Work*, pp. 418–423, 1999.

[224] R. Engels and C. Theusinger, “Using a data metric for preprocessing advice for data mining applications,” in *In Proceedings of the European Conference on Artificial Intelligence (ECAI-98*, pp. 430–434, John Wiley and Sons, 1998.

[225] M. Reif, F. Shafait, and A. Dengel, "Meta2-features: Providing meta-learners more information," in *35th German Conference on Artificial Intelligence*, Citeseer, 2012.

[226] J. W. Tukey, *Exploratory data analysis*, vol. 2. Pearson; 1 edition., 1977.

[227] H. Zhang, "Case retrieval strategy of distributed clustering algorithm based on min-cluster," *Boletin Tecnico, ISSN: 0376-723X*, vol. 55, no. 9, 2017.

[228] R. M. Esteves, T. Hacker, and C. Rong, "Competitive k-means, a new accurate and distributed k-means algorithm for large datasets," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 1, pp. 17–24, Dec 2013.

[229] J. Wu, *Advances in K-means Clustering: A Data Mining Thinking*. Springer Publishing Company, Incorporated, 2012.

[230] D. C. Corrales, J. C. Corrales, A. Sanchis, and A. Ledezma, "Sequential classifiers for network intrusion detection based on data selection process," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 001827–001832, Oct 2016.

[231] J. A. Hartigan and M. A. Wong, "A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[232] K. L. Spafford, J. S. Meredith, and J. S. Vetter, "Quartile and outlier detection on heterogeneous clusters using distributed radix sort," in *2011 IEEE International Conference on Cluster Computing*, pp. 412–419, Sept 2011.

[233] W. C. Krumbein, "The use of quartile measures in describing and comparing sediments," *American Journal of Science*, no. 188, pp. 98–111, 1936.

[234] A. El-Sayed, C. Montgomery, and S. Jenkins, "Utilization of quartile analysis in process control," in *Proceedings of ISSM2000. Ninth International Symposium on Semiconductor Manufacturing (IEEE Cat. No.00CH37130)*, pp. 442–445, 2000.

[235] J. De Leeuw and S. Pruzansky, "A new computational method to fit the weighted euclidean distance model," *Psychometrika*, vol. 43, no. 4, pp. 479–490, 1978.

[236] G. N. Lance and W. T. Williams, "Mixed-data classificatory programs i - agglomerative systems.," *Australian Computer Journal*, vol. 1, no. 1, pp. 15–20, 1967.

[237] S. Slade, “Case-based reasoning: A research paradigm,” *AI magazine*, vol. 12, no. 1, p. 42, 1991.

[238] P. H. S. Panahy, F. Sidi, L. S. Affendey, M. A. Jabar, H. Ibrahim, and A. Mustapha, “A framework to construct data quality dimensions relationships,” *Indian Journal of Science and Technology*, vol. 6, no. 5, pp. 4422–4431, 2013.

[239] M. Bovee, R. P. Srivastava, and B. Mak, “A conceptual framework and belief-function approach to assessing overall information quality,” *International journal of intelligent systems*, vol. 18, no. 1, pp. 51–74, 2003.

[240] D. P. Ballou and H. L. Pazer, “Modeling data and process quality in multi-input, multi-output information systems,” *Manage. Sci.*, vol. 31, pp. 150–162, Feb. 1985.

[241] D. Barone, F. Stella, and C. Batini, “Dependency discovery in data quality,” in *Advanced Information Systems Engineering* (B. Pernici, ed.), (Berlin, Heidelberg), pp. 53–67, Springer Berlin Heidelberg, 2010.

[242] R. L. De Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T COX, K. Forbus, *et al.*, “Retrieval, reuse, revision and retention in case-based reasoning,” *The Knowledge Engineering Review*, vol. 20, no. 3, pp. 215–240, 2005.

[243] Y.-B. Kang, S. Krishnaswamy, and A. Zaslavsky, “Retrieval in cbr using a combination of similarity and association knowledge,” in *Advanced Data Mining and Applications* (J. Tang, I. King, L. Chen, and J. Wang, eds.), (Berlin, Heidelberg), pp. 1–14, Springer Berlin Heidelberg, 2011.

[244] A. Ordoñez, H. Ordoñez, J. C. Corrales, C. Cobos, L. K. Wives, and L. H. Thom, “Grouping of business processes models based on an incremental clustering algorithm using fuzzy similarity and multimodal search,” *Expert Systems with Applications*, vol. 67, pp. 163 – 177, 2017.

[245] D. D. Lewis, *Representation and Learning in Information Retrieval.* PhD thesis, Amherst, MA, USA, 1992. UMI Order No. GAX92-19460.

[246] D. C. Corrales, J. E. Gomez, and J. C. Corrales, *Comparación estructural y linguistica de procesos de negocio semánticos.* Research and Innovation Book, 1st ed., 2012.

[247] C. Figueroa, D. C. Corrales, and J. C. Corrales, “Un enfoque multinivel para la recuperación de procesos de negocio,” *Revista Ingenierías Universidad de Medellín*, vol. 14, no. 26, 2015.

[248] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 202–207, AAAI Press, 1996.

[249] S. D. Vito, M. Piga, L. Martinotto, and G. D. Francia, "Co, no2 and nox urban pollution monitoring with on-field calibrated electronic nose by automatic bayesian regularization," *Sensors and Actuators B: Chemical*, vol. 143, no. 1, pp. 182 – 191, 2009.

[250] D. C. Corrales, A. F. Casas, A. Ledezma, and J. C. Corrales, "Two-level classifier ensembles for coffee rust estimation in colombian crops," *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, vol. 7, no. 3, pp. 41–59, 2016.

[251] Y. Guo, Y. Peng, and J. Hu, "Research on high creative application of case-based reasoning system on engineering design," *Computers in Industry*, vol. 64, no. 1, pp. 90 – 103, 2013.

[252] A. P. Engelbrecht and R. Brits, "A clustering approach to incremental learning for feedforward neural networks," in *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, vol. 3, pp. 2019–2024 vol.3, 2001.

[253] S. Young, I. Arel, T. P. Karnowski, and D. Rose, "A fast and stable incremental clustering algorithm," in *2010 Seventh International Conference on Information Technology: New Generations*, pp. 204–209, April 2010.

[254] R. Polikar, L. Upda, S. S. Upda, and V. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 31, pp. 497–508, Nov 2001.

[255] M. Kuhn, "Building predictive models in r using the caret package," *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008.

[256] P. Clements, D. Garlan, R. Little, R. Nord, and J. Stafford, "Documenting software architectures: Views and beyond," in *Proceedings of the 25th International Conference on Software Engineering*, ICSE '03, (Washington, DC, USA), pp. 740–741, IEEE Computer Society, 2003.

[257] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[258] A. Jena, “Apache jena,” *jena. apache. org [Online]. Available: http://jena. apache. org [Accessed: May. 6, 2018]*, p. 14, 2013.

[259] K. Chodorow, *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage.* ” O’Reilly Media, Inc.”, 2013.

[260] Maven, “Opencsv,” *opencsv.sourceforge.net [Online]. Available: http://opencsv.sourceforge.net/ [Accessed: May. 6, 2018]*, 2017.

[261] C. Apache, “Commons lang,” *commons.apache.org [Online]. Available: https://commons.apache.org/proper/commons-lang/ [Accessed: May. 6, 2018]*, 2017.

[262] S. Urbanek, “Package ’rserve’ manual,” *URL [https://cran.r-project.org/web/packages/Rserve/Rserve.pdf]*, 2012.

[263] D. Stekhoven, “Package ’missforest’ manual,” *URL [https://cran.r-project.org/web/packages/missForest/missForest.pdf]*, 2016.

[264] S. v. Buuren and K. Groothuis-Oudshoorn, “mice: Multivariate imputation by chained equations in r,” *Journal of statistical software*, pp. 1–68, 2010.

[265] Y. Hu, W. Murray, and Y. Shan, “Package ’rlof’ manual,” *URL [https://cran.r-project.org/web/packages/Rlof/Rlof.pdf]*, 2015.

[266] C. Hennig, “Package ’fpc’ manual,” *URL [https://cran.r-project.org/web/packages/fpc/fpc.pdf]*, 2018.

[267] P. Branco, R. Ribeiro, and L. Torgo, “Package ’ubl’ manual,” *URL [https://cran.r-project.org/web/packages/UBL/UBL.pdf]*, 2017.

[268] W. Siriseriwan, “Package ’smotefamily’ manual,” *URL [https://cran.r-project.org/web/packages/smotefamily/smotefamily.pdf]*, 2018.

[269] P. Romanski and L. Kotthoff, “Package ’fselector’ manual,” *URL [https://cran.r-project.org/web/packages/FSelector/FSelector.pdf]*, 2016.

Appendices

Universidad
del Cauca ®

uc3m | Universidad **Carlos III** de Madrid

ANEXOS

En convenio específico de cotutela: 23–32. 7 – 088 2015

UNIVERSIDAD DEL CAUCA
Programa de Doctorado en Ingeniería Telemática

UNIVERSIDAD CARLOS III DE MADRID
Programa de Doctorado en Ciencia y Tecnología Informática

Framework for Data Quality in Knowledge Discovery Tasks

AUTOR
David Camilo Corrales Muñoz

DIRECTOR
Dr. Juan Carlos Corrales
CO-DIRECTOR
Dr. Agapito Ledezma Espino

Noviembre 2018

A. Conceptual framework

A.1 Case study: office occupancy

The authors in [173] proposed a dataset for prediction of occupancy in an office room using six variables: temperature, humidity, humidity ratio, light CO_2 and the class occupancy status (0 for non-occupied, 1 for occupied). Three data sets were used, one for training (8143 instances), and two for testing the models (Test 1: 2665 instances and Test 2: 9752).

A.1.0.1 Outliers detection

The first step was to apply outliers detection task. We used Local Outlier Factor (LOF) and Tukey fences, then 872 potential outliers were found by Tukey fences. We considered potential outliers the instances with LOF among 0.808 (lower fence) and 1.297 (upper fence). After removing the potential outliers, 1600 instances represent that the room is occupied (Yes), and 5671 non-occupied (No).

A.1.0.2 Label correction

To correct the labels of the classes, we used Contradictory instances detection. The dataset has no contradictory instances.

A.1.0.3 Classes balancing

In balanced classes task, We used Synthetic Minority Over-sampling Technique (Smote) due the imbalance ratio of classes is 3.7. Thus, 4000 instances were added to the minority class. Figure A.1 shows the instance distribution per class for all subjects. Purple bars represent the imbalanced dataset, and blue bars the balanced dataset using Smote. Thus, 4000 instances were added to the minority class.

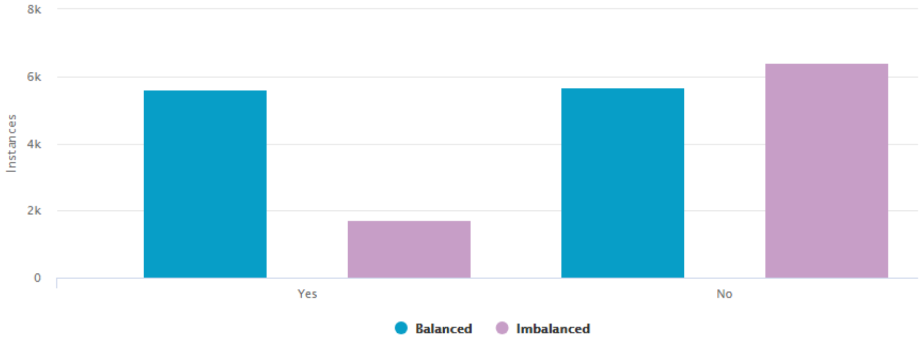


Figure A.1: Instance distribution per class: balanced vs imbalanced

A.1.0.4 Remove duplicate instances

To detect duplicate instances we used again Standard Duplicate Elimination. We removed 812 duplicate instances (809 non-occupied and 3 occupied).

A.1.0.5 Results

The authors in [173] used the classifiers: Random Forest (RF), Gradient Boosting Machines (GBM), Linear Discriminant Analysis (LDA) and Classification and Regression Trees (CART) with a CARET package available in R [255]. For the classifiers, we used the same experimental configuration proposed by the authors [173]. Table A.1 presents the accuracies for mentioned models with 10-fold cross-validation, once applied our approach, occupancy detection with original attributes and preprocessing attributes.

Table A.1: Results of dataset occupancy detection of an office room

Approach	**Model**	**Test 1**	**Test 2**
Our approach	RF	94.90	**99.25**
	GBM	94.78	**96.68**
	CART	**97.75**	**98.70**
	LDA	97.90	98.68
Occupancy detection (original attributes)	RF	95.05	97.16
	GBM	93.06	95.14
	CART	95.57	96.47
	LDA	97.90	98.76
Occupancy detection (preprocessing attributes)	RF	**95.53**	98.06
	GBM	**95.76**	96.10
	CART	94.52	96.52
	LDA	97.90	**99.33**

For Test 1, once applied our conceptual framework on training data, the accuracy of the models RF y GBM are 0.63 and 0.98 percentage points below of the best result of occupancy detection with preprocessing attributes. For CART model our conceptual framework obtained the highest *Accuracy* (97.75), while the three approaches obtained the same *Accuracy* for LDA (97.90).

For Test 2, our conceptual framework reaches the highest *Accuracy* in RF (99.25), GBM (96.68) and CART (98.70) models. The highest *Accuracy* for LDA model (99.33) is obtained by Occupancy detection with preprocessing attributes.

The good results obtained by occupancy detection with preprocessing (RF , GBM in Test 1 and LDA in Test 2) can be due to two new attributes included: the number of seconds from midnight for each day and Week status (weekend or a weekday).

A.2 Description of the dataset comments prediction in Facebook

The dataset for regression tasks was proposed in [160], which is oriented to the prediction of comments in a Facebook post. The dataset is composed of a data test with 10.120 instances and five training sets. The description of the attributes are presented in Table A.2.

Table A.2: Attributes description of the dataset comments prediction in Facebook. Attribute Index corresponds to number of the attribute, the Attribute Type is the category defined by the dataset authors [160], and the description represents the attribute definition.

Attribute Index	**Attribute Type**	**Description**
1	Likes	Number of likes in the post.
2	Page Checkin	Localization of the post visitors.
3	Page talking about	Total of activities such as comments, likes, shares and visitors.
4	Page Category	Post category source (place, institution, brand, etc).
5 - 29	Statistical measures	Minimum, maximum, average, median and standard deviation of essential features.
30	CC1	Comments before selected base date/time.
31	CC2	Comments in last 24 hours.
32	CC3	Comments between the last 48 and 24 hours.
33	CC4	Comments in the first 24 hours after the publication of post.
34	CC5	CC2 - CC3.
35	Base time	Selected time in order to simulate the scenario.
36	Post length	Number of characters that contains the post.
37	Post Share Count	Number of times that the post was shared.
38	Post Promotion Status	Binary attribute, the post is promoted or not.
39	H Local	Received comments per hour.
40-46	Post published weekday	Day of the week when the post was published.
47-53	Base DateTime weekday	Day of the week on selected base Date/Time.
54	Target Variable	Comments in the next H hours.

A.3 Results

The conceptual framework was tested through 48 datasets (28 datasets for classification and 20 for regression) of the UCI Repository of Machine Learning Databases [158] of the last twenty years (1998 - 2018). The process for testing the conceptual framework (CF) consists of three steps:

1. The UCI datasets are cleaned by our conceptual framework (CF).
2. The cleaned datasets by our conceptual framework (CF) are used to train the same algorithms proposed by authors of UCI datasets.
3. We compare the performance measures (i.e. for classification: textitPrecision, textitArea Under Curve and regression: textitMean Absolute Error) of the models trained with the datasets produced by the authors versus the models trained with the datasets processed by our conceptual framework.

Tables A.3, A.4, A.5, A.6, A.7, we present the results of the classifiers trained with the same algorithms proposed by authors of UCI datasets for classification tasks. Similarly, in Tables A.8, A.9, we present the results of the regression models.

A.3.1 Classification

Table A.3: Results of the classifiers processed by conceptual framework (CF)

Dataset	Model	Tool	Validation	Measure	Value
Anuran families calls	C4.5	Weka	Leave-one-out Cross-validation	Accuracy	0.976
Anuran families calls	K nearest neighbor	Weka	Leave-one-out Cross-validation	Accuracy	0.998
Anuran families calls	Support vector machine	Weka	Leave-one-out Cross-validation	Accuracy	0.905
Anuran species calls	C4.5	Weka	Leave-one-out Cross-validation	Accuracy	0.989
Anuran species calls	K nearest neighbor	Weka	Leave-one-out Cross-validation	Accuracy	0.998
Anuran species calls	Support vector machine	Weka	Leave-one-out Cross-validation	Accuracy	0.993
Autism in adolescent	Random forest	Weka	10 Cross validation	Accuracy	0.998
Autism in adolescent	C4.5	Weka	10 Cross validation	Accuracy	0.991
Autism in adolescent	REP tree	Weka	10 Cross validation	Accuracy	0.986
Autism in adult	Random forest	Weka	10 Cross validation	Accuracy	0.989
Autism in adult	C4.5	Weka	10 Cross validation	Accuracy	0.991
Autism in adult	REP tree	Weka	10 Cross validation	Accuracy	0.983
Autism in child	Random forest	Weka	10 Cross validation	Accuracy	0.997
Autism in child	C4.5	Weka	10 Cross validation	Accuracy	0.993
Autism in child	REP tree	Weka	10 Cross validation	Accuracy	0.99
Bank telemarketing	Decision tree	R - rminer	10 Cross validation	AUC	0.898
Bank telemarketing	Neural network	R - rminer	10 Cross validation	AUC	0.926
Bank telemarketing	Support vector machine	R - rminer	10 Cross validation	AUC	0.76
Breast tissue detection	Linear discriminant analysis	Weka	10 Cross validation	AUC	0.9221
Breast tissue detection	Support vector machine	Weka	10 Cross validation	AUC	0.864
Cardiotocography	Random forest	Weka	10 Cross validation	Accuracy	0.984
Cardiotocography	Random tree	Weka	10 Cross validation	Accuracy	0.962
Cardiotocography	C4.5	Weka	10 Cross validation	Accuracy	0.986

Table A.4: Results of the classifiers processed by conceptual framework (CF)

Dataset	Model	Tool	Validation	Measure	Value
Chemi. biodegradability	Support vector machine	Weka	10 Cross validation	AUC	0.867
Chemi. biodegradability	K nearest neighbor	Weka	10 Cross validation	AUC	0.888
Chemi. biodegradability	Linear discriminant analysis	Weka	10 Cross validation	AUC	0.932
Chemi. biodegradability	Multi layer perceptron	Weka	10 Cross validation	AUC	0.941
Chemi. biodegradability	Ada boost - C4.5	Weka	10 Cross validation	AUC	0.955
Chronic Kidney	C4.5	Weka	10 Cross validation	Accuracy	0.992
Chronic Kidney	Multi layer perceptron	Weka	10 Cross validation	Accuracy	0.989
Chronic Kidney	Support vector machine	Weka	10 Cross validation	Accuracy	0.991
Companies bankruptcy 1	C4.5	Weka	10 Cross validation	AUC - mean	0.77
Companies bankruptcy 1	Multi layer perceptron	Weka	10 Cross validation	AUC - mean	0.663
Companies bankruptcy 1	Support vector machine	Weka	10 Cross validation	AUC - mean	0.502
Companies bankruptcy 2	C4.5	Weka	10 Cross validation	AUC - mean	0.739
Companies bankruptcy 2	Multi layer perceptron	Weka	10 Cross validation	AUC - mean	0.517
Companies bankruptcy 2	Support vector machine	Weka	10 Cross validation	AUC - mean	0.502
Companies bankruptcy 3	C4.5	Weka	10 Cross validation	AUC - mean	0.805
Companies bankruptcy 3	Multi layer perceptron	Weka	10 Cross validation	AUC - mean	0.593
Companies bankruptcy 3	Support vector machine	Weka	10 Cross validation	AUC - mean	0.5
Companies bankruptcy 4	C4.5	Weka	10 Cross validation	AUC - mean	0.802
Companies bankruptcy 4	Multi layer perceptron	Weka	10 Cross validation	AUC - mean	0.68
Companies bankruptcy 4	Support vector machine	Weka	10 Cross validation	AUC - mean	0.501

Table A.5: Results of the classifiers processed by conceptual framework (CF)

Dataset	Model	Tool	Validation	Measure	Value
Companies bankruptcy 5	C4.5	Weka	10 Cross validation	AUC - mean	0.834
Companies bankruptcy 5	Multi layer perceptron	Weka	10 Cross validation	AUC - mean	0.835
Companies bankruptcy 5	Support vector machine	Weka	10 Cross validation	AUC - mean	0.522
Default of credit card	C4.5	Weka	10 Cross validation	AUC	0.834
Default of credit card	Multi layer perceptron	Weka	10 Cross validation	AUC	0.836
Default of credit card	Support vector machine	Weka	10 Cross validation	AUC	0.67
Human activity recog.	Support vector machine	Weka	data test	Accuracy	0.984
Office occupancy	Random forest	R - caret	data test	Accuracy	0.9925
Office occupancy	Gradient boosting machines	R - caret	data test	Accuracy	0.9668
Office occupancy	Classification and regression trees	R - caret	data test	Accuracy	0.987
Office occupancy	Linear discriminant analysis	R - caret	data test	Accuracy	0.9868
Ozone level 1 hour	C4.5	Weka	10 Cross validation	Accuracy	0.941
Ozone level 1 hour	Bagging C4.5	Weka	10 Cross validation	Accuracy	0.963
Ozone level 8 hours	C4.5	Weka	10 Cross validation	Accuracy	0.913
Ozone level 8 hours	Bagging C4.5	Weka	10 Cross validation	Accuracy	0.927
Phishing detection	C4.5	Weka	10 Cross validation	Accuracy	0.838
Phishing detection	REP tree	Weka	10 Cross validation	Accuracy	0.822
Phishing detection	Random forest	Weka	10 Cross validation	Accuracy	0.828
Phishing websites	Multi layer perceptron	Weka	10 Cross validation	AUC	0.98
Phishing websites	Radial basis function network	Weka	10 Cross validation	AUC	0.854
Phishing websites	Voted perceptron	Weka	10 Cross validation	AUC	0.923

Table A.6: Results of the classifiers processed by conceptual framework (CF)

Dataset	Model	Tool	Validation	Measure	Value
Physical activity - 1	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 1	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 1	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 1	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Physical activity - 2	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 2	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 2	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 2	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Physical activity - 3	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 3	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 3	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 3	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Physical activity - 4	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 4	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 4	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 4	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Physical activity - 5	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 5	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 5	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 5	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997

Table A.7: Results of the classifiers processed by conceptual framework (CF)

Dataset	Model	Tool	Validation	Measure	Value
Physical activity - 6	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 6	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 6	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 6	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Physical activity - 7	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 7	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 7	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 7	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Physical activity - 8	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 8	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 8	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 8	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Physical activity - 9	C4.5	Weka	Leave-one-subject-out	Accuracy	0.993
Physical activity - 9	Boosted C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.999
Physical activity - 9	Bagging C4.5 decision tree	Weka	Leave-one-subject-out	Accuracy	0.96
Physical activity - 9	K nearest neighbor	Weka	Leave-one-subject-out	Accuracy	0.997
Risk cervical cancer	Support vector machine	Weka	Hold out: 80-20	Accuracy	0.932
Seismic hazard predic.	C4.5	Weka	10 Cross validation	Accuracy	0.937
Seismic hazard predic.	Naive bayes	Weka	10 Cross validation	Accuracy	0.881
Seismic hazard predic.	Random forest	Weka	10 Cross validation	Accuracy	0.946
Vertebral column diagn.	Multi layer perceptron	Weka	Hold out: 80-20	Accuracy	0.855
Vertebral column diagn.	Support vector machine	Weka	Hold out: 80-20	Accuracy	0.774
Vertebral column injury	Multi layer perceptron	Weka	Hold out: 80-20	Accuracy	0.872
Vertebral column injury	Support vector machine	Weka	Hold out: 80-20	Accuracy	0.882
Voice rehabilitation	Support vector machine	Weka	10 Cross validation	Accuracy	0.881

A.3.2 Regression

Table A.8: Results of the regression models processed by conceptual framework (CF)

Dataset	Model	Tool	Validation	Measure	Value
Air Pollution Benzene	Random forest	Weka	data test	MAE	0.7481
Airfoil Self Noise	Random forest	Weka	10 cross validation	MAE	1.2036
Airfoil Self Noise	Multi layer perceptron	Weka	10 cross validation	MAE	3.6212
Airfoil Self Noise	Radial basis function	Weka	10 cross validation	MAE	5.5973
Airfoil Self Noise	Support vector regression	Weka	10 cross validation	MAE	3.6778
Beijing PM 2.5 pollution	Support vector regression	Weka	10 cross validation	MAE	21.412
Comments prediction in FB - 1	Multi layer perceptron	Weka	data test	MAE	34.55
Comments prediction in FB - 1	Radial basis function	Weka	data test	MAE	33.09
Comments prediction in FB - 1	REP tree	Weka	data test	MAE	34.08
Comments prediction in FB - 1	M5P tree	Weka	data test	MAE	35.53
Comments prediction in FB - 2	Multi layer perceptron	Weka	data test	MAE	31.31
Comments prediction in FB - 2	Radial basis function	Weka	data test	MAE	31.85
Comments prediction in FB - 2	REP tree	Weka	data test	MAE	30.22
Comments prediction in FB - 2	M5P tree	Weka	data test	MAE	30.32
Comments prediction in FB - 3	Multi layer perceptron	Weka	data test	MAE	49.19
Comments prediction in FB - 3	Radial basis function	Weka	data test	MAE	31.12
Comments prediction in FB - 3	REP tree	Weka	data test	MAE	28.41
Comments prediction in FB - 3	M5P tree	Weka	data test	MAE	32.68
Comments prediction in FB - 4	Multi layer perceptron	Weka	data test	MAE	48.59
Comments prediction in FB - 4	Radial basis function	Weka	data test	MAE	29.81
Comments prediction in FB - 4	REP tree	Weka	data test	MAE	27.89
Comments prediction in FB - 4	M5P tree	Weka	data test	MAE	50.77
Comments prediction in FB - 5	Multi layer perceptron	Weka	data test	MAE	43.47
Comments prediction in FB - 5	Radial basis function	Weka	data test	MAE	29.69
Comments prediction in FB - 5	REP tree	Weka	data test	MAE	29.33
Comments prediction in FB - 5	M5P tree	Weka	data test	MAE	32.59

Table A.9: Results of the regression models processed by conceptual framework (CF)

Dataset	Model	Tool	Validation	Measure	Value
Compressor decay	Support vector regression	Weka	10 cross-validation	MAE	0.0057
Energy use of appliances	Random forest	R - caret	data test	MAE	12.138
Feedback Blogs Prediction	M5P	Weka	data test	MAE	5.8802
Feedback Blogs Prediction	REP tree	Weka	data test	MAE	5.7057
Feedback Blogs Prediction	K nearest neighbor	Weka	data test	MAE	8.0627
Feedback Blogs Prediction	Linear regression	Weka	data test	MAE	9.333
Feedback Blogs Prediction	Multi layer perceptron	Weka	data test	MAE	7.9462
Forest Fires	Random forest	Weka	10 cross-validation	MAE	17.696
I-Dinning room temperature	Multi layer perceptron	Weka	10 cross validation	MAE	0.4794
I-Dinning room temperature	Radial basis function	Weka	10 cross validation	MAE	2.262
I-Dinning room temperature	Linear regression	Weka	10 cross validation	MAE	0.6144
I-Room temperature	Multi layer perceptron	Weka	10 cross validation	MAE	0.4302
I-Room temperature	Radial basis function	Weka	10 cross validation	MAE	2.235
I-Room temperature	Linear regression	Weka	10 cross validation	MAE	0.6157
II-Dinning room temperature	Multi layer perceptron	Weka	10 cross validation	MAE	0.3454
II-Dinning room temperature	Radial basis function	Weka	10 cross validation	MAE	2.0767
II-Dinning room temperature	Linear regression	Weka	10 cross validation	MAE	0.5971
II-Room temperature	Multi layer perceptron	Weka	10 cross validation	MAE	0.3241
II-Room temperature	Radial basis function	Weka	10 cross validation	MAE	2.0654
II-Room temperature	Linear regression	Weka	10 cross validation	MAE	0.5491
Posts in Facebook pages	Support vector regression	Weka	10 cross validation	MAE	25.26
Rental Bikes Daily	Linear regression	Weka	data test	MAE	5E-05
Rental Bikes Daily	REP tree	Weka	data test	MAE	29.312
Rental Bikes Hourly	Linear regression	Weka	data test	MAE	1E-05
Rental Bikes Hourly	REP tree	Weka	data test	MAE	10.653
Turbine decay	Support vector regression	Weka	10 cross-validation	MAE	0.0031

A.4 Datasets similarity

A.4.1 Classification

Table A.10: Dataset 1: Anuran families calls

Meta-features	Authors	CF	Similarity	Measure
Attributes	22	20	0.9524	Canberra
Instances	7195	8829	0.8980	Canberra
Data Dimensionality	0.0031	0.0023	0.8511	Canberra
Mean Skewness	0.8688	0.7032	0.8947	Canberra
Mean Kurtosis	0.0395	0.0352	0.9420	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9620	1.0000	0.9620	Euclidean
Imbalance Ratio	1.0000	1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.6678 %	

Table A.11: Dataset 2: Anuran species calls

Meta-features	Authors	CF	Similarity	Measure
Attributes	22	20	0.9524	Canberra
Instances	7195	7189	0.9996	Canberra
Data Dimensionality	0.0031	0.0028	0.9528	Canberra
Mean Skewness	0.8688	0.5163	0.7455	Canberra
Mean Kurtosis	0.0395	0.0258	0.7906	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9992	0.9992	1.0000	Euclidean

Table A.11: Dataset 2: Anuran species calls

Meta-features	Authors	CF	Similarity	Measure
Imbalance Ratio	1.0000	1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.2726 %	

Table A.12: Dataset 3: Autism in adolescent

Meta-features	Authors	CF	Similarity	Measure
Attributes	20	20	1.0000	Canberra
Instances	104	123	0.9163	Canberra
Data Dimensionality	0.1923	0.1626	0.9163	Canberra
Mean Skewness	0.7206	0.5662	0.8800	Canberra
Mean Kurtosis	0.0601	0.0472	0.8800	Arithmetic
Mean Entropy	0.6483	0.6198	0.9715	Euclidean
Mutual Information Class	0.0494	0.0518	0.9766	Arithmetic
Mean Abs Correlation	0.2361	0.2398	0.9963	Euclidean
Equiv Num-Features	19.5853	19.3158	0.9931	Canberra
Noise Signal	12.1245	10.9726	0.9501	Canberra
Missing Values	0.0050	0.0000	0.9950	Euclidean
Duplicate Instances	0.0100	0.0000	0.9900	Euclidean
Class Entropy	0.9675	1.0000	0.9675	Euclidean
Imbalance Ratio	1.0000	1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.2170 %	

Table A.13: Dataset 4: Autism in adult

Meta-features	Authors	CF	Similarity	Measure
Attributes	20	19	0.9744	Canberra
Instances	704	696	0.9943	Canberra
Data Dimensionality	0.0284	0.0273	0.9801	Canberra
Mean Skewness	1.5696	0.4749	0.4645	Canberra
Mean Kurtosis	0.1308	0.0396	0.4645	Arithmetic
Mean Entropy	0.5153	0.3446	0.8292	Euclidean
Mutual Information Class	0.0277	0.0941	0.4551	Arithmetic
Mean Abs Correlation	0.1753	0.4428	0.7325	Euclidean

Table A.13: Dataset 4: Autism in adult

Meta-features	Authors	CF	Similarity	Measure
Equiv Num-Features	30.2825	4.9619	0.2816	Canberra
Noise Signal	17.5942	2.6624	0.2629	Canberra
Missing Values	0.0130	0.0000	0.9870	Euclidean
Duplicate Instances	0.0070	0.0000	0.9930	Euclidean
Class Entropy	0.8393	0.4668	0.6276	Euclidean
Imbalance Ratio	2.0000	1.0000	0.6667	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	69.6044 %	

Table A.14: Dataset 5: Autism in child

Meta-features	Authors	CF	Similarity	Measure
Attributes	20	20	1.0000	Canberra
Instances	292	290	0.9966	Canberra
Data Dimensionality	0.0685	0.0690	0.9966	Canberra
Mean Skewness	0.5424	0.5412	0.9989	Canberra
Mean Kurtosis	0.0452	0.0451	0.9989	Arithmetic
Mean Entropy	0.5809	0.5579	0.9770	Euclidean
Mutual Information Class	0.0267	0.0285	0.9670	Arithmetic
Mean Abs Correlation	0.0856	0.0872	0.9983	Euclidean
Equiv Num-Features	37.4865	35.0907	0.9670	Canberra
Noise Signal	20.7962	18.5947	0.9441	Canberra
Missing Values	0.0150	0.0000	0.9850	Euclidean
Duplicate Instances	0.0070	0.0000	0.9930	Euclidean
Class Entropy	0.9992	0.9991	1.0000	Euclidean
Imbalance Ratio	1.0000	1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	98.8154 %	

Table A.15: Dataset 6: Breast tissue detection

Meta-features	Authors	CF	Similarity	Measure
Attributes	9	9	1.0000	Canberra
Instances	106	105	0.9953	Canberra
Data Dimensionality	0.0849	0.0857	0.9953	Canberra

Table A.15: Dataset 6: Breast tissue detection

Meta-features	Authors	CF	Similarity	Measure
Mean Skewness	2.2535	2.2405	0.9971	Canberra
Mean Kurtosis	0.2504	0.2489	0.9971	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0090	0.0000	0.9910	Euclidean
Class Entropy	0.9921	0.9914	0.9993	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	99.8330 %	

Table A.16: Dataset 7: Cardiotocography

Meta-features	Authors	CF	Similarity	Measure
Attributes	22	22	1.0000	Canberra
Instances	2126	2289	0.9631	Canberra
Data Dimensionality	0.0103	0.0096	0.9631	Canberra
Mean Skewness	0.8303	0.8198	0.9936	Canberra
Mean Kurtosis	0.0377	0.0373	0.9936	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0060	0.0000	0.9940	Euclidean
Class Entropy	0.6147	0.7154	0.8992	Euclidean
Imbalance Ratio	-1.0000	-1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	98.7111 %	

Table A.17: Dataset 8: Default of credit card

Meta-features	Authors	CF	Similarity	Measure
Attributes	23	17	0.8500	Canberra
Instances	30000	36598	0.9009	Canberra
Data Dimensionality	0.0008	0.0005	0.7546	Canberra
Mean Skewness	5.3246	6.4634	0.9034	Canberra
Mean Kurtosis	0.2315	0.3802	0.7569	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0010	0.0003	0.9993	Euclidean
Class Entropy	0.7624	0.9447	0.8177	Euclidean
Imbalance Ratio	3.0000	1.0000	0.5000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	89.8854 %	

Table A.18: Dataset 9: Human activity recog.

Meta-features	Authors	CF	Similarity	Measure
Attributes	561	553	0.9928	Canberra
Instances	4252	4219	0.9961	Canberra
Data Dimensionality	0.1319	0.1311	0.9967	Canberra
Mean Skewness	2.0895	1.8764	0.9463	Canberra
Mean Kurtosis	0.0037	0.0034	0.9534	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0070	0.0000	0.9930	Euclidean
Class Entropy	0.9949	0.9950	0.9999	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	99.1878 %	

Table A.19: Dataset 10: Ozone level 1 hour

Meta-features	Authors	CF	Similarity	Measure
Attributes	72	71	0.9930	Canberra
Instances	2536	2021	0.8870	Canberra
Data Dimensionality	0.0284	0.0351	0.8939	Canberra
Mean Skewness	0.7153	0.7715	0.9622	Canberra
Mean Kurtosis	0.0099	0.0109	0.9552	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0810	0.0000	0.9190	Euclidean
Duplicate Instances	0.0030	0.0000	0.9970	Euclidean
Class Entropy	0.1883	0.5084	0.6799	Euclidean
Imbalance Ratio	33.0000	7.0000	0.3500	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	90.9143 %	

Table A.20: Dataset 11: Ozone level 8 hours

Meta-features	Authors	CF	Similarity	Measure
Attributes	72	72	1.0000	Canberra
Instances	2534	2233	0.9369	Canberra
Data Dimensionality	0.0284	0.0322	0.9369	Canberra
Mean Skewness	0.7156	0.8428	0.9183	Canberra
Mean Kurtosis	0.0099	0.0117	0.9183	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0810	0.0000	0.9190	Euclidean
Duplicate Instances	0.0030	0.0000	0.9970	Euclidean
Class Entropy	0.3398	0.7768	0.5630	Euclidean
Imbalance Ratio	14.0000	3.0000	0.3529	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	90.2822 %	

Table A.21: Dataset 12: Phishing detection

Meta-features	Authors	CF	Similarity	Measure
Attributes	9	8	0.9412	Canberra
Instances	1353	885	0.7909	Canberra
Data Dimensionality	0.0067	0.0090	0.8478	Canberra
Mean Skewness	0.5511	0.3231	0.7392	Canberra
Mean Kurtosis	0.0612	0.0404	0.7949	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.4650	0.1020	0.6370	Euclidean
Class Entropy	0.8215	0.9839	0.8376	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	90.5908 %	

Table A.22: Dataset 13: Office occupancy

Meta-features	Authors	CF	Similarity	Measure
Attributes	5	4	0.8889	Canberra
Instances	8143	10733	0.8628	Canberra
Data Dimensionality	0.0006	0.0004	0.7554	Canberra
Mean Skewness	0.9914	0.4990	0.6696	Canberra
Mean Kurtosis	0.1983	0.1248	0.7724	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.1060	0.0000	0.8940	Euclidean
Class Entropy	0.7459	0.9992	0.7467	Euclidean
Imbalance Ratio	3.0000	1.0000	0.5000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	87.2656 %	

Table A.23: Dataset 14: Phishing websites

Meta-features	Authors	CF	Similarity	Measure
Attributes	30	30	1.0000	Canberra
Instances	11055	5849	0.6920	Canberra
Data Dimensionality	0.0027	0.0051	0.6920	Canberra
Mean Skewness	1.4718	1.3344	0.9511	Canberra
Mean Kurtosis	0.0491	0.0445	0.9511	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.4710	0.0000	0.5290	Euclidean
Class Entropy	0.9906	0.9992	0.9914	Euclidean
Imbalance Ratio	1.0000	1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	92.0436 %	

Table A.24: Dataset 15: Chronic Kidney

Meta-features	Authors	CF	Similarity	Measure
Attributes	24	22	0.9565	Canberra
Instances	400	550	0.8421	Canberra
Data Dimensionality	0.0600	0.0400	0.8000	Canberra
Mean Skewness	2.7887	2.5355	0.9525	Canberra
Mean Kurtosis	0.1992	0.1950	0.9895	Arithmetic
Mean Entropy	0.5156	0.5440	0.9716	Euclidean
Mutual Information Class	0.1167	0.1276	0.9553	Arithmetic
Mean Abs Correlation	0.9892	0.9913	0.9979	Euclidean
Equiv Num-Features	8.1784	7.7891	0.9756	Canberra
Noise Signal	3.4181	3.2628	0.9768	Canberra
Missing Values	0.1010	0.0000	0.8990	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9544	0.9940	0.9604	Euclidean
Imbalance Ratio	1.0000	1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	95.1807 %	

Table A.25: Dataset 16.1: Physical activity - subject 1

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	376417	260769	0.8185	Canberra
Data Dimensionality	0.0001	0.0002	0.8372	Canberra
Mean Skewness	0.6067	0.5534	0.9541	Canberra
Mean Kurtosis	0.0114	0.0109	0.9733	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0200	0.0000	0.9800	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.8871	0.9936	0.8936	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.2494 %	

Table A.26: Dataset 16.2: Physical activity - subject 2

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	447000	280375	0.7709	Canberra
Data Dimensionality	0.0001	0.0002	0.7892	Canberra
Mean Skewness	0.6994	0.6857	0.9901	Canberra
Mean Kurtosis	0.0132	0.0134	0.9907	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0210	0.0000	0.9790	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.8230	0.9938	0.8292	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	95.5327 %	

Table A.27: Dataset 16.3: Physical activity - subject 3

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	252833	174075	0.8155	Canberra
Data Dimensionality	0.0002	0.0003	0.8342	Canberra
Mean Skewness	0.6267	0.5823	0.9633	Canberra
Mean Kurtosis	0.0118	0.0114	0.9825	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0180	0.0000	0.9820	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9205	0.9897	0.9308	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.5937 %	

Table A.28: Dataset 16.4: Physical activity - subject 4

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	329576	223754	0.8088	Canberra
Data Dimensionality	0.0002	0.0002	0.8274	Canberra
Mean Skewness	0.4767	0.4662	0.9890	Canberra
Mean Kurtosis	0.0090	0.0091	0.9918	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0200	0.0000	0.9800	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.8893	0.9540	0.9353	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.7528 %	

Table A.29: Dataset 16.5: Physical activity - subject 5

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	374783	298895	0.8874	Canberra
Data Dimensionality	0.0001	0.0002	0.9064	Canberra
Mean Skewness	0.6388	0.6546	0.9878	Canberra
Mean Kurtosis	0.0121	0.0128	0.9686	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0200	0.0000	0.9800	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9165	0.9958	0.9207	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	97.5434 %	

Table A.30: Dataset 16.6: Physical activity - subject 6

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	361817	264943	0.8454	Canberra
Data Dimensionality	0.0001	0.0002	0.8643	Canberra
Mean Skewness	0.9219	0.9280	0.9967	Canberra
Mean Kurtosis	0.0174	0.0182	0.9775	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0190	0.0000	0.9810	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.8771	0.9589	0.9182	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	97.0925 %	

Table A.31: Dataset 16.7: Physical activity - subject 7

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	313599	263632	0.9134	Canberra
Data Dimensionality	0.0002	0.0002	0.9326	Canberra
Mean Skewness	0.7268	0.7776	0.9662	Canberra
Mean Kurtosis	0.0137	0.0152	0.9470	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0200	0.0000	0.9800	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9166	0.9934	0.9232	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	97.6215 %	

Table A.32: Dataset 16.8: Physical activity - subject 8

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	408031	309190	0.8622	Canberra
Data Dimensionality	0.0001	0.0002	0.8811	Canberra
Mean Skewness	0.8895	0.9035	0.9922	Canberra
Mean Kurtosis	0.0168	0.0177	0.9730	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0210	0.0000	0.9790	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.8583	0.9963	0.8620	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.8679 %	

Table A.33: Dataset 16.9: Physical activity - subject 9

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	51	0.9808	Canberra
Instances	8477	5810	0.8133	Canberra
Data Dimensionality	0.0063	0.0088	0.8320	Canberra
Mean Skewness	0.9958	0.7153	0.8361	Canberra
Mean Kurtosis	0.0188	0.0140	0.8549	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0190	0.0000	0.9810	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.8050	0.9877	0.8174	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	91.5211 %	

Table A.34: Dataset 17: Companies bankruptcy 1

Meta-features	Authors	CF	Similarity	Measure
Attributes	64	55	0.9244	Canberra
Instances	7027	7471	0.9694	Canberra
Data Dimensionality	0.0091	0.0074	0.8940	Canberra
Mean Skewness	52.7715	49.4735	0.9677	Canberra
Mean Kurtosis	0.8246	0.8995	0.9565	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0130	0.0000	0.9870	Euclidean
Duplicate Instances	0.0120	0.0040	0.9920	Euclidean
Class Entropy	0.2357	0.4890	0.7467	Euclidean
Imbalance Ratio	24.0000	8.0000	0.5000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	92.9179 %	

Table A.35: Dataset 18: Companies bankruptcy 2

Meta-features	Authors	CF	Similarity	Measure
Attributes	64	56	0.9333	Canberra
Instances	10173	10734	0.9732	Canberra
Data Dimensionality	0.0063	0.0052	0.9067	Canberra
Mean Skewness	76.2130	70.3467	0.9600	Canberra
Mean Kurtosis	1.1908	1.2562	0.9733	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0180	0.0000	0.9820	Euclidean
Duplicate Instances	0.0090	0.0000	0.9914	Euclidean
Class Entropy	0.2392	0.4969	0.7423	Euclidean
Imbalance Ratio	24.0000	8.0000	0.5000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	93.0805 %	

Table A.36: Dataset 19: Companies bankruptcy 3

Meta-features	Authors	CF	Similarity	Measure
Attributes	64	57	0.9421	Canberra
Instances	10503	11418	0.9583	Canberra
Data Dimensionality	0.0061	0.0050	0.9006	Canberra
Mean Skewness	64.3523	59.8952	0.9641	Canberra
Mean Kurtosis	1.0055	1.0508	0.9780	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0140	0.0000	0.9860	Euclidean
Duplicate Instances	0.0080	0.0002	0.9922	Euclidean
Class Entropy	0.2741	0.5564	0.7177	Euclidean
Imbalance Ratio	20.0000	6.0000	0.4615	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	92.6704 %	

Table A.37: Dataset 20: Companies bankruptcy 4

Meta-features	Authors	CF	Similarity	Measure
Attributes	64	56	0.9333	Canberra
Instances	9792	10694	0.9560	Canberra
Data Dimensionality	0.0065	0.0052	0.8896	Canberra
Mean Skewness	58.2686	59.4352	0.9901	Canberra
Mean Kurtosis	0.9104	1.0613	0.9235	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0140	0.0000	0.9860	Euclidean
Duplicate Instances	0.0080	0.0020	0.9940	Euclidean
Class Entropy	0.2973	0.5932	0.7041	Euclidean
Imbalance Ratio	18.0000	5.0000	0.4348	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	92.0760 %	

Table A.38: Dataset 21: Companies bankruptcy 5

Meta-features	Authors	CF	Similarity	Measure
Attributes	64	58	0.9508	Canberra
Instances	5910	6326	0.9660	Canberra
Data Dimensionality	0.0108	0.0092	0.9170	Canberra
Mean Skewness	53.6974	40.1245	0.8553	Canberra
Mean Kurtosis	0.8390	0.6918	0.9038	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0120	0.0000	0.9880	Euclidean
Duplicate Instances	0.0100	0.0002	0.9902	Euclidean
Class Entropy	0.3636	0.6694	0.6942	Euclidean
Imbalance Ratio	13.0000	4.0000	0.4706	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	91.5728 %	

Table A.39: Dataset 22: Bank telemarketing

Meta-features	Authors	CF	Similarity	Measure
Attributes	16	13	0.8966	Canberra
Instances	45211	50500	0.9447	Canberra
Data Dimensionality	0.0004	0.0003	0.8422	Canberra
Mean Skewness	8.8059	8.2608	0.9681	Canberra
Mean Kurtosis	1.2580	1.3768	0.9549	Arithmetic
Mean Entropy	0.6974	0.7509	0.9465	Euclidean
Mutual Information Class	0.0103	0.0217	0.6446	Arithmetic
Mean Abs Correlation	0.1603	0.2286	0.9317	Euclidean
Equiv Num-Features	50.4283	34.1068	0.8069	Canberra
Noise Signal	66.5482	33.5869	0.6708	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	0.9998	Euclidean
Class Entropy	0.5206	0.7405	0.7802	Euclidean
Imbalance Ratio	7.0000	3.0000	0.6000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	86.5797 %	

Table A.40: Dataset 23: Chemi. biodegradability

Meta-features	Authors	CF	Similarity	Measure
Attributes	39	36	0.9600	Canberra
Instances	1379	1379	1.0000	Canberra
Data Dimensionality	0.0283	0.0261	0.9600	Canberra
Mean Skewness	3.3744	3.3443	0.9955	Canberra
Mean Kurtosis	0.0865	0.0653	0.8603	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9997	0.9997	1.0000	Euclidean
Imbalance Ratio	1.0000	1.0000	1.0000	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	98.5054 %	

Table A.41: Dataset 24: Risk cervical cancer

Meta-features	Authors	CF	Similarity	Measure
Attributes	35	31	0.9394	Canberra
Instances	858	887	0.9834	Canberra
Data Dimensionality	0.0408	0.0349	0.9229	Canberra
Mean Skewness	7.1880	7.8921	0.9533	Canberra
Mean Kurtosis	0.2054	0.2546	0.8930	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.1170	0.0000	0.8830	Euclidean
Duplicate Instances	0.0270	0.0000	0.9730	Euclidean
Class Entropy	0.3435	0.5376	0.8059	Euclidean
Imbalance Ratio	14.0000	7.0000	0.6667	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	93.4701 %	

Table A.42: Dataset 25: Seismic hazard predic.

Meta-features	Authors	CF	Similarity	Measure
Attributes	18	15	0.9091	Canberra
Instances	2584	3258	0.8846	Canberra
Data Dimensionality	0.0070	0.0046	0.7959	Canberra
Mean Skewness	4.3700	5.1362	0.9194	Canberra
Mean Kurtosis	0.3121	0.4669	0.8013	Arithmetic
Mean Entropy	0.7160	0.6848	0.9689	Euclidean
Mutual Information Class	0.0042	0.0183	0.3746	Arithmetic
Mean Abs Correlation	0.1759	0.3466	0.8293	Euclidean
Equiv Num-Features	82.7728	45.1391	0.7058	Canberra
Noise Signal	168.3181	36.3306	0.3551	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0020	0.0000	0.9980	Euclidean
Class Entropy	0.3500	0.8281	0.5219	Euclidean
Imbalance Ratio	14.0000	2.0000	0.2500	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	75.4255 %	

Table A.43: Dataset 26: Vertebral column diagn.

Meta-features	Authors	CF	Similarity	Measure
Attributes	6	6	1.0000	Canberra
Instances	310	410	0.8611	Canberra
Data Dimensionality	0.0194	0.0146	0.8611	Canberra
Mean Skewness	1.1749	1.3538	0.9293	Canberra
Mean Kurtosis	0.1958	0.2256	0.9293	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9072	0.9996	0.9076	Euclidean
Imbalance Ratio	2.0000	1.0000	0.6667	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	94.3667 %	

Table A.44: Dataset 27: Vertebral column injury

Meta-features	Authors	CF	Similarity	Measure
Attributes	6	6	1.0000	Canberra
Instances	310	370	0.9118	Canberra
Data Dimensionality	0.0194	0.0162	0.9118	Canberra
Mean Skewness	1.1749	1.2848	0.9553	Canberra
Mean Kurtosis	0.1958	0.2141	0.9553	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9413	0.9874	0.9538	Euclidean
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	97.9201 %	

Table A.45: Dataset 28: Voice rehabilitation

Meta-features	Authors	CF	Similarity	Measure
Attributes	310	56	0.3060	Canberra
Instances	126	168	0.8571	Canberra
Data Dimensionality	2.4603	0.3333	0.2386	Canberra
Mean Skewness	3.5840	3.5044	0.9888	Canberra
Mean Kurtosis	0.0116	0.0626	0.3119	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mutual Information Class	0.0000	0.0000	1.0000	Arithmetic
Mean Abs Correlation	0.0000	0.0000	1.0000	Euclidean
Equiv Num-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Class Entropy	0.9183	1.0000	0.9183	Euclidean
Imbalance Ratio	2.0000	1.0000	0.6667	Canberra
MissingValues Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	81.9160 %	

A.4.2 Regression

Table A.46: Dataset 1: Airfoil Self Noise

Meta-features	Authors	CF	Similarity	Measure
Attributes	5	4	0.8889	Canberra
Instances	1503	1503	1.0000	Canberra
Data Dimensionality	0.0033	0.0027	0.8889	Canberra
Mean Skewness	1.0433	1.0851	0.9804	Canberra
Mean Kurtosis	0.2087	0.2143	0.9866	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.2442	0.3202	0.9240	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	2.6829	2.6829	1.0000	Canberra
Skewness Of Class	-0.4185	-0.4185	1.0000	Canberra
Outliers Of Class	0.0030	0.0030	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	97.7911 %	

Table A.47: Dataset 2: Beijing PM 2.5 pollution

Meta-features	Authors	CF	Similarity	Measure
Attributes	11	5	0.6250	Canberra
Instances	43824	43824	1.0000	Canberra
Data Dimensionality	0.0003	0.0001	0.6250	Canberra
Mean Skewness	3.5873	1.1779	0.4944	Canberra
Mean Kurtosis	0.3587	0.2945	0.9016	Arithmetic
Mean Entropy	0.9450	0.9450	1.0000	Euclidean
Mean Abs Correlation	0.0772	0.1380	0.9392	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0040	0.0000	0.9960	Euclidean
Duplicate Instances	0.0000	0.0003	0.9997	Euclidean
Kurtosis Of Class	7.7682	7.7081	0.9961	Canberra
Skewness Of Class	1.8022	1.7969	0.9985	Canberra
Outliers Of Class	0.0410	0.0430	0.9980	Euclidean
MissingValues Of Class	0.0470	0.0000	0.9530	Euclidean
		Similarity	90.1770 %	

Table A.48: Dataset 3: Comments prediction in FB – 1

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	35	0.7955	Canberra
Instances	40949	40932	0.9998	Canberra
Data Dimensionality	0.0013	0.0009	0.7957	Canberra
Mean Skewness	16.8759	18.2719	0.9603	Canberra
Mean Kurtosis	0.3184	0.5221	0.7577	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.1598	0.1033	0.9435	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0002	0.0000	0.9998	Euclidean
Kurtosis Of Class	301.4432	301.3462	0.9998	Canberra
Skewness Of Class	14.2928	14.2908	0.9999	Canberra
Outliers Of Class	0.1410	0.1410	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	95.0129 %	

Table A.49: Dataset 4: Comments prediction in FB – 2

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	34	0.7816	Canberra
Instances	81312	81285	0.9998	Canberra
Data Dimensionality	0.0007	0.0004	0.7818	Canberra
Mean Skewness	15.4166	16.2602	0.9734	Canberra
Mean Kurtosis	0.2909	0.4782	0.7564	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.1650	0.1014	0.9363	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0003	0.0000	0.9997	Euclidean
Kurtosis Of Class	481.7807	481.6554	0.9999	Canberra
Skewness Of Class	17.1575	17.1555	0.9999	Canberra
Outliers Of Class	0.1410	0.1410	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	94.8592 %	

Table A.50: Dataset 5: Comments prediction in FB – 3

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	34	0.7816	Canberra
Instances	121098	121033	0.9997	Canberra
Data Dimensionality	0.0004	0.0003	0.7819	Canberra
Mean Skewness	15.9805	17.9073	0.9431	Canberra
Mean Kurtosis	0.3015	0.5267	0.7281	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.1698	0.1019	0.9321	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0005	0.0000	0.9995	Euclidean
Kurtosis Of Class	369.5285	369.4117	0.9998	Canberra
Skewness Of Class	14.8109	14.8091	0.9999	Canberra
Outliers Of Class	0.1400	0.1400	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	94.4388 %	

Table A.51: Dataset 6: Comments prediction in FB – 4

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	34	0.7816	Canberra
Instances	160424	160325	0.9997	Canberra
Data Dimensionality	0.0003	0.0002	0.7819	Canberra
Mean Skewness	15.1443	15.3472	0.9933	Canberra
Mean Kurtosis	0.2857	0.4514	0.7753	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.1673	0.1035	0.9362	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0010	0.0000	0.9990	Euclidean
Kurtosis Of Class	462.0992	462.0800	1.0000	Canberra
Skewness Of Class	16.1195	16.1204	1.0000	Canberra
Outliers Of Class	0.1400	0.1400	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	95.1130 %	

Table A.52: Dataset 7: Comments prediction in FB – 5

Meta-features	Authors	CF	Similarity	Measure
Attributes	53	33	0.7674	Canberra
Instances	199030	198881	0.9996	Canberra
Data Dimensionality	0.0003	0.0002	0.7678	Canberra
Mean Skewness	14.0258	13.6872	0.9878	Canberra
Mean Kurtosis	0.2646	0.4148	0.7790	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.1697	0.0977	0.9280	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0010	0.0000	0.9990	Euclidean
Kurtosis Of Class	355.4998	356.1682	0.9991	Canberra
Skewness Of Class	14.9703	14.9854	0.9995	Canberra
Outliers Of Class	0.1410	0.1410	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	94.8482 %	

Table A.53: Dataset 8: Compressor decay

Meta-features	Authors	CF	Similarity	Measure
Attributes	16	13	0.8966	Canberra
Instances	11934	11934	1.0000	Canberra
Data Dimensionality	0.0013	0.0011	0.8966	Canberra
Mean Skewness	0.5061	0.5609	0.9487	Canberra
Mean Kurtosis	0.0316	0.0431	0.8461	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.0131	0.0160	0.9970	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	1.7991	1.7991	1.0000	Canberra
Skewness Of Class	0.0000	0.0000	1.0000	Canberra
Outliers Of Class	0.0000	0.0000	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	97.2328 %	

Table A.54: Dataset 9: Turbine decay

Meta-features	Authors	CF	Similarity	Measure
Attributes	16	14	0.9333	Canberra
Instances	11934	11934	1.0000	Canberra
Data Dimensionality	0.0013	0.0012	0.9333	Canberra
Mean Skewness	0.5061	0.5045	0.9984	Canberra
Mean Kurtosis	0.0316	0.0315	0.9984	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.0084	0.0082	0.9998	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	1.7964	1.7964	1.0000	Canberra
Skewness Of Class	0.0000	0.0000	1.0000	Canberra
Outliers Of Class	0.0000	0.0000	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	99.0884 %	

Table A.55: Dataset 10: Rental Bikes Hourly

Meta-features	Authors	CF	Similarity	Measure
Attributes	14	13	0.9630	Canberra
Instances	8645	8642	0.9998	Canberra
Data Dimensionality	0.0016	0.0015	0.9631	Canberra
Mean Skewness	0.8864	0.8863	0.9999	Canberra
Mean Kurtosis	0.0633	0.0633	0.9999	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.2819	0.2817	0.9998	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0001	0.0000	0.9999	Euclidean
Kurtosis Of Class	3.7587	3.7442	0.9981	Canberra
Skewness Of Class	1.1314	1.1278	0.9984	Canberra
Outliers Of Class	0.0250	0.0250	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	99.4798 %	

Table A.56: Dataset 11: Air Pollution Benzene

Meta-features	Authors	CF	Similarity	Measure
Attributes	12	8	0.8000	Canberra
Instances	5646	5605	0.9964	Canberra
Data Dimensionality	0.0021	0.0014	0.8035	Canberra
Mean Skewness	0.7745	0.5456	0.8266	Canberra
Mean Kurtosis	0.0645	0.0682	0.9725	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.4693	0.1732	0.7039	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.1380	0.0000	0.8620	Euclidean
Duplicate Instances	0.0050	0.0190	0.9860	Euclidean
Kurtosis Of Class	33.1315	44.7055	0.8513	Canberra
Skewness Of Class	-5.5173	-6.3788	0.9276	Canberra
Outliers Of Class	0.0520	0.0460	0.9940	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	91.4915 %	

Table A.57: Dataset 12: Rental Bikes Daily

Meta-features	Authors	CF	Similarity	Measure
Attributes	13	12	0.9600	Canberra
Instances	365	365	1.0000	Canberra
Data Dimensionality	0.0356	0.0329	0.9600	Canberra
Mean Skewness	0.7737	0.7679	0.9962	Canberra
Mean Kurtosis	0.0595	0.0589	0.9945	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.3725	0.3756	0.9970	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	1.9152	1.9152	1.0000	Canberra
Skewness Of Class	-0.3592	-0.3592	1.0000	Canberra
Outliers Of Class	0.0000	0.0000	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	99.3847 %	

Table A.58: Dataset 13: Energy use of appliances

Meta-features	Authors	CF	Similarity	Measure
Attributes	30	23	0.8679	Canberra
Instances	14803	14803	1.0000	Canberra
Data Dimensionality	0.0020	0.0016	0.8679	Canberra
Mean Skewness	0.4979	0.5645	0.9373	Canberra
Mean Kurtosis	0.0178	0.0257	0.8187	Arithmetic
Mean Entropy	0.9248	0.9998	0.9249	Euclidean
Mean Abs Correlation	0.0677	0.0825	0.9853	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	16.0681	16.0681	1.0000	Canberra
Skewness Of Class	3.3272	3.3272	1.0000	Canberra
Outliers Of Class	0.1100	0.1100	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.0135 %	

Table A.59: Dataset 14: Posts in Facebook pages

Meta-features	Authors	CF	Similarity	Measure
Attributes	18	14	0.8750	Canberra
Instances	500	499	0.9990	Canberra
Data Dimensionality	0.0360	0.0281	0.8760	Canberra
Mean Skewness	5.2178	4.7695	0.9551	Canberra
Mean Kurtosis	0.3069	0.3669	0.9110	Arithmetic
Mean Entropy	0.3970	0.4221	0.9749	Euclidean
Mean Abs Correlation	0.3485	0.4463	0.9021	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0010	0.0000	0.9990	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	14.2225	14.6572	0.9849	Canberra
Skewness Of Class	2.9827	3.0075	0.9959	Canberra
Outliers Of Class	0.1180	0.1180	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.4864 %	

Table A.60: Dataset 15: Feedback Blogs Prediction

Meta-features	Authors	CF	Similarity	Measure
Attributes	280	126	0.6207	Canberra
Instances	52397	49203	0.9686	Canberra
Data Dimensionality	0.0053	0.0026	0.6479	Canberra
Mean Skewness	25.8402	7.3540	0.4431	Canberra
Mean Kurtosis	0.0923	0.0584	0.7748	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.0699	0.1502	0.9196	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0610	0.0000	0.9510	Euclidean
Kurtosis Of Class	235.2954	230.1784	0.9890	Canberra
Skewness Of Class	12.6913	12.6214	0.9972	Canberra
Outliers Of Class	0.1950	0.1880	0.9930	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	88.6999 %	

Table A.61: Dataset 16: Forest Fires

Meta-features	Authors	CF	Similarity	Measure
Attributes	12	6	0.6667	Canberra
Instances	517	513	0.9961	Canberra
Data Dimensionality	0.0232	0.0117	0.6701	Canberra
Mean Skewness	3.2700	2.2062	0.8057	Canberra
Mean Kurtosis	0.3270	0.4412	0.8513	Arithmetic
Mean Entropy	0.8333	0.6748	0.8415	Euclidean
Mean Abs Correlation	0.0472	0.0542	0.9930	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0080	0.0120	0.9960	Euclidean
Kurtosis Of Class	195.2566	193.8489	0.9964	Canberra
Skewness Of Class	12.8096	12.7647	0.9982	Canberra
Outliers Of Class	0.1220	0.1210	0.9990	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	92.0939 %	

Table A.62: Dataset 17: I-Room temperature

Meta-features	Authors	CF	Similarity	Measure
Attributes	20	14	0.8235	Canberra
Instances	2764	2764	1.0000	Canberra
Data Dimensionality	0.0072	0.0051	0.8235	Canberra
Mean Skewness	1.6531	1.7560	0.9698	Canberra
Mean Kurtosis	0.0827	0.1254	0.7944	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.2156	0.2979	0.9177	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	2.8512	2.8512	1.0000	Canberra
Skewness Of Class	-0.3655	-0.3655	1.0000	Canberra
Outliers Of Class	0.0040	0.0040	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	95.5266 %	

Table A.63: Dataset 18: II-Room temperature

Meta-features	Authors	CF	Similarity	Measure
Attributes	20	16	0.8889	Canberra
Instances	1373	1373	1.0000	Canberra
Data Dimensionality	0.0146	0.0117	0.8889	Canberra
Mean Skewness	1.1734	1.0632	0.9508	Canberra
Mean Kurtosis	0.0587	0.0665	0.9378	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.2782	0.3445	0.9337	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	2.3672	2.3672	1.0000	Canberra
Skewness Of Class	-0.0891	-0.0891	1.0000	Canberra
Outliers Of Class	0.0000	0.0000	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	97.3332 %	

Table A.64: Dataset 19: I-Dinning room temperature

Meta-features	Authors	CF	Similarity	Measure
Attributes	20	13	0.7879	Canberra
Instances	2764	2764	1.0000	Canberra
Data Dimensionality	0.0072	0.0047	0.7879	Canberra
Mean Skewness	1.6531	1.7726	0.9651	Canberra
Mean Kurtosis	0.0827	0.1364	0.7548	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.2126	0.3163	0.8962	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	2.9034	2.9034	1.0000	Canberra
Skewness Of Class	-0.3835	-0.3835	1.0000	Canberra
Outliers Of Class	0.0040	0.0040	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	94.6128 %	

Table A.65: Dataset 20: II-Dinning room temperature

Meta-features	Authors	CF	Similarity	Measure
Attributes	20	15	0.8571	Canberra
Instances	1373	1373	1.0000	Canberra
Data Dimensionality	0.0146	0.0109	0.8571	Canberra
Mean Skewness	1.1734	1.1148	0.9744	Canberra
Mean Kurtosis	0.0587	0.0743	0.8823	Arithmetic
Mean Entropy	0.0000	0.0000	1.0000	Euclidean
Mean Abs Correlation	0.2656	0.3448	0.9208	Euclidean
Equiv Numb-Features	0.0000	0.0000	1.0000	Canberra
Noise Signal	0.0000	0.0000	1.0000	Canberra
Missing Values	0.0000	0.0000	1.0000	Euclidean
Duplicate Instances	0.0000	0.0000	1.0000	Euclidean
Kurtosis Of Class	2.3809	2.3809	1.0000	Canberra
Skewness Of Class	-0.0740	-0.0740	1.0000	Canberra
Outliers Of Class	0.0000	0.0000	1.0000	Euclidean
MissingValues Of Class	0.0000	0.0000	1.0000	Euclidean
		Similarity	96.6119 %	

B. Retrieval mechanism

In this Appendix, we present the retrieved cases by the clustering and quartile filters. In case of clustering, the k–means was applied for 2, 3, 4, 5, 6 and 7 clusters.

B.1 Panel of Judges

The panel of judges scores (0 - 100%) the similarity between a query case against all cases of the case-base.

B.1.1 Evaluations of Judge 1

B.1.1.1 Classification

Query 1: Autism spectrum disorder in children

The similarity results of the Query 1 are presented in Table B.1. In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/2.Ivan_Lopez/Query1_classification_ILopez.xlsx`

Query 2: Portuguese bank telemarketing

The similarity results of the Query 2 are presented in Table B.2. In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/2.Ivan_Lopez/Query2_classification_ILopez.xlsx`

Query 3: Income prediction

The similarity results of the Query 3 are presented in Table B.3. In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/2.Ivan_Lopez/Query3_classification_ILopez.xlsx`

Table B.1: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 1	17.9710255865
CASE 2	18.335232913
CASE 3	19.5766586857
CASE 4	19.924796932
CASE 5	19.9386942325
CASE 6	24.166420532
CASE 7	22.8240310566
CASE 8	24.9752815218
CASE 9	23.9324570382
CASE 10	24.2444503546
CASE 11	22.9169148562
CASE 12	23.8289580106
CASE 13	22.5008715666
CASE 14	29.2993677813
CASE 15	25.6392259572
CASE 16	43.9771202825
CASE 17	44.3488914738
CASE 18	44.4427016484
CASE 19	29.1942811938
CASE 20	100
CASE 21	78.0914957238
CASE 22	76.3895586787
CASE 23	62.424035953
CASE 24	35.0125561206
CASE 25	19.0204446537
CASE 26	22.6821556686
CASE 27	23.5247811497
CASE 28	23.2940097072
CASE 29	23.1192855718

Table B.1: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 30	39.4945298855
CASE 31	32.3746609524
CASE 32	20.194005498
CASE 33	22.1220865902
CASE 34	36.9381840869
CASE 35	41.0176564912
CASE 36	20.0964993963

Table B.2: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 1	22.5567008721
CASE 2	23.1759768082
CASE 3	25.312361611
CASE 4	26.598026951
CASE 5	30.4131385674
CASE 6	10.046595728
CASE 7	10.053084711
CASE 8	11.2601418838
CASE 9	10.3476263263
CASE 10	9.9392226093
CASE 11	10.31322196
CASE 12	10.4595110629
CASE 13	10.1297677042
CASE 14	15.4079482833
CASE 15	32.1198850789
CASE 16	21.2685997182
CASE 17	21.1288535273
CASE 18	63.3526329771
CASE 19	33.7071648214
CASE 20	36.7697005369
CASE 21	33.7642096663
CASE 22	36.1370044774
CASE 23	28.8278918928
CASE 24	26.3640737559
CASE 25	23.3482844609
CASE 26	21.7457523994

Table B.2: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 27	20.976590693
CASE 28	15.3981429162
CASE 29	30.5855659176
CASE 30	20.1007727897
CASE 31	22.8287845857
CASE 32	17.2971466524
CASE 33	26.8449028334
CASE 34	17.9840263105
CASE 35	55.2610310556
CASE 36	11.1828496055

Table B.3: Query 3: Income prediction

Name	Similarity (%)
CASE 1	15.4326355839
CASE 2	15.2525672927
CASE 3	16.4305413128
CASE 4	17.0119928537
CASE 5	17.9109444792
CASE 6	15.8105883707
CASE 7	16.4223596857
CASE 8	16.6508918997
CASE 9	15.8240258863
CASE 10	15.5954458139
CASE 11	16.6224130689
CASE 12	16.1660414175
CASE 13	16.5504806298
CASE 14	30.7342500207
CASE 15	46.0106103948
CASE 16	31.0335812149
CASE 17	30.7249852933
CASE 18	63.9094664048
CASE 19	51.2327442769
CASE 20	49.048463443
CASE 21	57.2206536562
CASE 22	59.4615163792
CASE 23	48.6903689081

Table B.3: Query 3: Income prediction

Name	Similarity (%)
CASE 24	40.7027679152
CASE 25	32.3847907921
CASE 26	31.7885491001
CASE 27	25.2124594604
CASE 28	29.328382583
CASE 29	19.8978320094
CASE 30	30.590829982
CASE 31	30.1458354891
CASE 32	12.8688877522
CASE 33	16.0055542475
CASE 34	30.292794558
CASE 35	48.0737660738
CASE 36	18.8502631605

B.1.1.2 Regression

The similarity results of the Query 1 are presented in Table B.4.

Query 1: Air pollution benzene estimation

Table B.4: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 1	38.7202263015
CASE 2	36.9660677695
CASE 3	36.8609874957
CASE 4	36.8688951975
CASE 5	37.0401757346
CASE 6	50.9524053607
CASE 7	50.9022910837
CASE 8	58.0579763466
CASE 9	63.0618743725
CASE 10	38.7728131924
CASE 11	34.2356706692
CASE 12	100
CASE 13	38.6284563435
CASE 14	47.4785392138
CASE 15	37.5999823467

Table B.4: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 16	35.4253404768
CASE 17	49.8780831847
CASE 18	50.5012632631
CASE 19	50.0358320406
CASE 20	50.485496346

In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/2.Ivan_Lopez/Query1_regression_ILopez.xlsx`

Query 2: Rental bikes hourly

The similarity results of the Query 2 are presented in Table B.5.

Table B.5: Query 2: Rental bikes hourly

Name	Similarity (%)
CASE 1	45.0728601613
CASE 2	43.0439431133
CASE 3	42.9406838904
CASE 4	43.5468732158
CASE 5	43.5583504153
CASE 6	61.5554497956
CASE 7	61.4633699384
CASE 8	65.1011568145
CASE 9	93.0828159609
CASE 10	39.1107190089
CASE 11	42.4284147438
CASE 12	64.5478377238
CASE 13	42.8380361079
CASE 14	58.8835811283
CASE 15	40.121796457
CASE 16	39.2465230341
CASE 17	62.3219414296
CASE 18	63.7051200987
CASE 19	62.3022921692
CASE 20	63.6211994034

In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/2.Ivan_Lopez/Query2_regression_ILopez.xlsx`

Query 3: Coffee rust

The similarity results of the Query 3 are presented in Table B.6.

Table B.6: Query 3: Coffee rust

Name	Similarity (%)
CASE 1	26.827156313
CASE 2	26.5278777821
CASE 3	26.5561129345
CASE 4	26.5494059447
CASE 5	26.6734449046
CASE 6	42.2535242954
CASE 7	42.0566323939
CASE 8	46.5861902054
CASE 9	53.8713700164
CASE 10	40.9499276558
CASE 11	52.702874895
CASE 12	41.2126686879
CASE 13	54.334459616
CASE 14	35.3898610991
CASE 15	54.7792975445
CASE 16	22.1065568902
CASE 17	48.0515017511
CASE 18	46.9961063505
CASE 19	47.9263615083
CASE 20	46.8702874165

In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/2.Ivan_Lopez/Query3_regression_ILopez.xlsx`

B.1.2 Evaluations of Judge 2

B.1.2.1 Classification

Query 1: Autism spectrum disorder in children

The similarity results of the Query 1 are presented in Table B.7. In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/3.Juan_Martinez/Query1_classification_JPMartinez.xlsx`

Query 2: Portuguese bank telemarketing

The similarity results of the Query 2 are presented in Table B.8. In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/3.Juan_Martinez/Query2_classification_JPMartinez.xlsx`

Query 3: Income prediction

The similarity results of the Query 3 are presented in Table B.9. In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/3.Juan_Martinez/Query3_classification_JPMartinez.xlsx`

Table B.7: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 1	22.1
CASE 2	22.3
CASE 3	22.5
CASE 4	21.85
CASE 5	22.2
CASE 6	21.75
CASE 7	25
CASE 8	26.3

Table B.7: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 9	22.4
CASE 10	23.7
CASE 11	24.05
CASE 12	24.7
CASE 13	19.5
CASE 14	30
CASE 15	31
CASE 16	39.15
CASE 17	39.15
CASE 18	38.65
CASE 19	26.1
CASE 20	100
CASE 21	57.85
CASE 22	69.05
CASE 23	57.9
CASE 24	32.3
CASE 25	29.05
CASE 26	33.95
CASE 27	31.9
CASE 28	27.6
CASE 29	25.65
CASE 30	35
CASE 31	34.1
CASE 32	22.9
CASE 33	22.9
CASE 34	25.4
CASE 35	31.6
CASE 36	21.05

Table B.8: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 1	20.95
CASE 2	21.69
CASE 3	22.94
CASE 4	23.69
CASE 5	25.39

Table B.8: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 6	11.05
CASE 7	12.98
CASE 8	12.51
CASE 9	10.93
CASE 10	10.95
CASE 11	10.52
CASE 12	11.32
CASE 13	10.14
CASE 14	13.84
CASE 15	30.59
CASE 16	22.39
CASE 17	15.79
CASE 18	60.54
CASE 19	30.99
CASE 20	33.93
CASE 21	30.24
CASE 22	31.44
CASE 23	26.51
CASE 24	25.02
CASE 25	24.56
CASE 26	19.75
CASE 27	20.28
CASE 28	19.08
CASE 29	20.65
CASE 30	23.42
CASE 31	21.60
CASE 32	14.69
CASE 33	15.62
CASE 34	20.46
CASE 35	46.61
CASE 36	13.99

Table B.9: Query 3: Income prediction

Name	Similarity (%)
CASE 1	18.42
CASE 2	17.63

Table B.9: Query 3: Income prediction

Name	Similarity (%)
CASE 3	17.93
CASE 4	18.99
CASE 5	18.87
CASE 6	15.86
CASE 7	16.08
CASE 8	17.46
CASE 9	16.18
CASE 10	16.36
CASE 11	16.76
CASE 12	16.59
CASE 13	16.48
CASE 14	21.59
CASE 15	38.60
CASE 16	29.30
CASE 17	29.10
CASE 18	63.29
CASE 19	45.97
CASE 20	45.92
CASE 21	57.48
CASE 22	53.90
CASE 23	46.82
CASE 24	34.33
CASE 25	31.95
CASE 26	30.63
CASE 27	25.10
CASE 28	27.27
CASE 29	19.70
CASE 30	29.50
CASE 31	28.52
CASE 32	13.81
CASE 33	15.05
CASE 34	30.49
CASE 35	49.84
CASE 36	21.06

B.1.2.2 Regression

Query 1: Air pollution benzene estimation

The similarity results of the Query 1 are presented in Table B.10.

Table B.10: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 1	46.9765334873
CASE 2	43.7318991687
CASE 3	39.7384513522
CASE 4	42.7562660955
CASE 5	43.4359069813
CASE 6	54.4797932821
CASE 7	54.4102689382
CASE 8	60.2232134079
CASE 9	71.4407584705
CASE 10	41.849643368
CASE 11	44.0203821759
CASE 12	100
CASE 13	34.3069857115
CASE 14	52.0782186498
CASE 15	44.778317248
CASE 16	44.1858392913
CASE 17	54.084815539
CASE 18	54.6662404102
CASE 19	54.9206486317
CASE 20	54.7026024636

In addition, the evaluation form of the Query 1 is located in:

http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/3.Juan_Martinez/Query1_regression_JPMartinez.xlsx

Query 2: Rental bikes hourly

The similarity results of the Query 2 are presented in Table B.11.

Table B.11: Query 2: Rental bikes hourly

Name	Similarity (%)
CASE 1	51.062699818
CASE 2	47.6472775849
CASE 3	46.7392558421
CASE 4	46.5049914863
CASE 5	46.4084447277
CASE 6	63.8172853595
CASE 7	63.6940326089
CASE 8	66.797536243
CASE 9	95.1680271947
CASE 10	40.9643228424
CASE 11	48.5548074471
CASE 12	70.5896723425
CASE 13	48.3090615438
CASE 14	64.2465472966
CASE 15	43.4826229306
CASE 16	45.1373520405
CASE 17	64.8266817809
CASE 18	67.9300094201
CASE 19	64.6786271389
CASE 20	67.7747844928

In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/3.Juan_Martinez/Query2_regression_JPMartinez.xlsx`

Query 3: Coffee rust

The similarity results of the Query 3 are presented in Table B.12.

Table B.12: Query 3: Coffee rust

Name	Similarity (%)
CASE 1	32.050947417
CASE 2	31.8088734783
CASE 3	31.70312192
CASE 4	32.0771959278
CASE 5	32.1860598852

Table B.12: Query 3: Coffee rust

Name	Similarity (%)
CASE 6	43.2774742466
CASE 7	43.0052754554
CASE 8	52.8443451964
CASE 9	52.9205741844
CASE 10	46.6674658182
CASE 11	49.2505050068
CASE 12	43.1327515246
CASE 13	54.1520550101
CASE 14	40.2995569354
CASE 15	52.640056196
CASE 16	29.6591992482
CASE 17	51.0717684436
CASE 18	46.2452910172
CASE 19	50.9145280538
CASE 20	49.8858919784

In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/3.Juan_Martinez/Query3_regression_JPMartinez.xlsx`

B.1.3 Evaluations of Judge 3

B.1.3.1 Classification

Query 1: Autism spectrum disorder in children

The similarity results of the Query 1 are presented in Table B.13. In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/4.Julian_Plazas/Query1_classification_JEPlazas.xlsx`

Query 2: Portuguese bank telemarketing

The similarity results of the Query 2 are presented in Table B.14. In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/4.Julian_Plazas/Query2_classification_JEPlazas.xlsx`

Query 3: Income prediction

The similarity results of the Query 3 are presented in Table B.15. In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/4.Julian_Plazas/Query3_classification_JEPlazas.xlsx`

Table B.13: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 1	42.7438704955
CASE 2	42.4330503877
CASE 3	42.8550238605
CASE 4	43.1670454319
CASE 5	44.2148834929
CASE 6	43.7163106051
CASE 7	42.5126182991
CASE 8	43.9121757047
CASE 9	43.2362300139
CASE 10	43.7715089707
CASE 11	42.3690365979
CASE 12	43.2969243665
CASE 13	42.2289390597
CASE 14	48.5858629797
CASE 15	45.2398027059
CASE 16	60.658237798
CASE 17	60.999807067
CASE 18	55.3269438122
CASE 19	44.8227364676
CASE 20	100
CASE 21	86.5483365817
CASE 22	80.8438007905

Table B.13: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 23	65.2538450717
CASE 24	60.2625331854
CASE 25	53.8568076817
CASE 26	49.2759823571
CASE 27	48.9047527884
CASE 28	48.3436969115
CASE 29	41.4786345711
CASE 30	62.3397315391
CASE 31	49.9669962377
CASE 32	51.7605055888
CASE 33	53.3431458876
CASE 34	58.3943695138
CASE 35	51.9929036356
CASE 36	45.1513963928

Table B.14: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 1	46.5784313678
CASE 2	47.0663156562
CASE 3	47.7855327005
CASE 4	48.2297574495
CASE 5	49.7338688673
CASE 6	33.2320383976
CASE 7	33.5294061772
CASE 8	34.0522414924
CASE 9	33.468856766
CASE 10	32.9752915834
CASE 11	33.5905632222
CASE 12	33.4680678855
CASE 13	33.4715167044
CASE 14	39.1798585475
CASE 15	51.4461002302
CASE 16	38.52205981
CASE 17	38.1795691906
CASE 18	69.7589066534
CASE 19	46.2635912873

Table B.14: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 20	50.1496669369
CASE 21	49.6797927553
CASE 22	51.8170479943
CASE 23	43.5840547375
CASE 24	38.5953428288
CASE 25	32.1564061941
CASE 26	33.0107626205
CASE 27	35.3918850868
CASE 28	46.8219329713
CASE 29	57.6502608116
CASE 30	44.260108828
CASE 31	30.408383716
CASE 32	41.085879835
CASE 33	44.9372825727
CASE 34	37.7307613976
CASE 35	60.4317185229
CASE 36	30.1702296433

Table B.15: Query 3: Income prediction

Name	Similarity (%)
CASE 1	43.46537826
CASE 2	42.3106657
CASE 3	43.43377302
CASE 4	44.06733422
CASE 5	44.95149092
CASE 6	37.5226488
CASE 7	38.08536783
CASE 8	38.13935926
CASE 9	37.42692674
CASE 10	37.32840311
CASE 11	38.49549807
CASE 12	37.90044027
CASE 13	38.36995604
CASE 14	48.20253824
CASE 15	52.75648959
CASE 16	47.64135762

Table B.15: Query 3: Income prediction

Name	Similarity (%)
CASE 17	47.298867
CASE 18	66.69437415
CASE 19	57.8062945
CASE 20	69.40057526
CASE 21	74.55124678
CASE 22	76.86501998
CASE 23	62.70651606
CASE 24	57.52366023
CASE 25	49.5255075
CASE 26	48.03636321
CASE 27	39.95352943
CASE 28	50.26915342
CASE 29	42.64086011
CASE 30	52.24043462
CASE 31	40.31770937
CASE 32	45.64167944
CASE 33	47.60275698
CASE 34	52.88137022
CASE 35	60.15097805
CASE 36	42.35436255

B.1.3.2 Regression

Query 1: Air pollution benzene estimation

The similarity results of the Query 1 are presented in Table B.16.

Table B.16: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 1	55.95964535
CASE 2	54.61348082
CASE 3	53.87045687
CASE 4	53.75226856
CASE 5	53.73324549
CASE 6	73.9599155
CASE 7	73.93666444
CASE 8	71.23793738

Table B.16: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 9	76.50926951
CASE 10	45.63354677
CASE 11	45.7681002
CASE 12	100
CASE 13	57.30809795
CASE 14	65.8282941
CASE 15	45.59622316
CASE 16	63.95271434
CASE 17	74.9933178
CASE 18	74.07670005
CASE 19	75.06713642
CASE 20	74.0679207

In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/4.Julian_Plazas/Query1_regression_JEPlazas.xlsx`

Query 2: Rental bikes hourly

The similarity results of the Query 2 are presented in Table B.17.

Table B.17: Query 2: Rental bikes hourly

Name	Similarity (%)
CASE 1	61.40651155
CASE 2	59.22255053
CASE 3	58.45526517
CASE 4	58.14439468
CASE 5	58.15067264
CASE 6	71.3371913
CASE 7	71.31083937
CASE 8	80.26895769
CASE 9	87.52254516
CASE 10	47.03352746
CASE 11	51.89105334
CASE 12	72.55899021
CASE 13	53.51325536
CASE 14	69.51775133

Table B.17: Query 2: Rental bikes hourly

Name	Similarity (%)
CASE 15	51.00695991
CASE 16	60.86188686
CASE 17	72.25822014
CASE 18	71.44578924
CASE 19	72.17748943
CASE 20	71.47123246

In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/4.Julian_Plazas/Query2_regression_JEPlazas.xlsx`

Query 3: Coffee rust

The similarity results of the Query 3 are presented in Table B.18.

Table B.18: Query 3: Coffee rust

Name	Similarity (%)
CASE 1	42.94268109
CASE 2	42.9135129
CASE 3	42.90619533
CASE 4	42.94500495
CASE 5	43.11324959
CASE 6	52.66240205
CASE 7	52.63741381
CASE 8	56.46890855
CASE 9	61.5119606
CASE 10	55.77631575
CASE 11	58.26926151
CASE 12	51.99400562
CASE 13	62.02653669
CASE 14	48.79449041
CASE 15	65.01861991
CASE 16	43.93021511
CASE 17	56.610533
CASE 18	52.40880895
CASE 19	56.48330746
CASE 20	52.41572448

In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/4.Julian_Plazas/Query3_regression_JEPlazas.xlsx`

B.1.4 Evaluations of Judge 4

B.1.4.1 Classification

Query 1: Autism spectrum disorder in children

The similarity results of the Query 1 are presented in Table B.19. In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/5.Sebastian_Rojas/Query1_classification_JSRojas.xlsx`

Query 2: Portuguese bank telemarketing

The similarity results of the Query 2 are presented in Table B.20. In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/5.Sebastian_Rojas/Query2_classification_JSRojas.xlsx`

Query 3: Income prediction

The similarity results of the Query 3 are presented in Table B.21. In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/5.Sebastian_Rojas/Query3_classification_JSRojas.xlsx`

Table B.19: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 1	2
CASE 2	2
CASE 3	2

Table B.19: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 4	2
CASE 5	2
CASE 6	0
CASE 7	0
CASE 8	0
CASE 9	0
CASE 10	0
CASE 11	0
CASE 12	0
CASE 13	0
CASE 14	2
CASE 15	2
CASE 16	5
CASE 17	10
CASE 18	15
CASE 19	2
CASE 20	100
CASE 21	85
CASE 22	80
CASE 23	50
CASE 24	2
CASE 25	2
CASE 26	2
CASE 27	0
CASE 28	0
CASE 29	2
CASE 30	5
CASE 31	0
CASE 32	2
CASE 33	2
CASE 34	5
CASE 35	2
CASE 36	2

Table B.20: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 1	2
CASE 2	2
CASE 3	5
CASE 4	5
CASE 5	5
CASE 6	0
CASE 7	0
CASE 8	0
CASE 9	0
CASE 10	0
CASE 11	0
CASE 12	0
CASE 13	0
CASE 14	2
CASE 15	2
CASE 16	2
CASE 17	2
CASE 18	20
CASE 19	2
CASE 20	5
CASE 21	5
CASE 22	2
CASE 23	5
CASE 24	2
CASE 25	0
CASE 26	0
CASE 27	0
CASE 28	2
CASE 29	2
CASE 30	2
CASE 31	0
CASE 32	2
CASE 33	2
CASE 34	2
CASE 35	20
CASE 36	0

Table B.21: Query 3: Income prediction

Name	Similarity (%)
CASE 1	0
CASE 2	0
CASE 3	0
CASE 4	0
CASE 5	0
CASE 6	0
CASE 7	5
CASE 8	0
CASE 9	0
CASE 10	0
CASE 11	0
CASE 12	0
CASE 13	0
CASE 14	5
CASE 15	2
CASE 16	0
CASE 17	0
CASE 18	25
CASE 19	2
CASE 20	0
CASE 21	15
CASE 22	8
CASE 23	5
CASE 24	0
CASE 25	0
CASE 26	0
CASE 27	0
CASE 28	0
CASE 29	0
CASE 30	0
CASE 31	0
CASE 32	0
CASE 33	0
CASE 34	5
CASE 35	45
CASE 36	0

B.1.4.2 Regression

Query 1: Air pollution benzene estimation:

The similarity results of the Query 1 are presented in Table B.22.

Table B.22: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 1	0
CASE 2	0
CASE 3	0
CASE 4	2
CASE 5	0
CASE 6	25
CASE 7	22
CASE 8	60
CASE 9	55
CASE 10	5
CASE 11	0
CASE 12	100
CASE 13	5
CASE 14	5
CASE 15	0
CASE 16	0
CASE 17	5
CASE 18	5
CASE 19	8
CASE 20	8

In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/5.Sebastian_Rojas/Query1_regression_JSRojas.xlsx`

Query 2: Rental bikes hourly

The similarity results of the Query 2 are presented in Table B.23.

Table B.23: Query 2: Rental bikes hourly

Name	Similarity (%)
CASE 1	2
CASE 2	0
CASE 3	0
CASE 4	0
CASE 5	0
CASE 6	25
CASE 7	25
CASE 8	82
CASE 9	90
CASE 10	5
CASE 11	5
CASE 12	30
CASE 13	5
CASE 14	50
CASE 15	2
CASE 16	0
CASE 17	30
CASE 18	30
CASE 19	40
CASE 20	40

In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/5.Sebastian_Rojas/Query2_regression_JSRojas.xlsx`

Query 3: Coffee rust

The similarity results of the Query 3 are presented in Table B.24.

Table B.24: Query 3: Coffee rust

Name	Similarity (%)
CASE 1	5
CASE 2	2
CASE 3	2
CASE 4	2
CASE 5	2
CASE 6	0

Table B.24: Query 3: Coffee rust

Name	Similarity (%)
CASE 7	0
CASE 8	10
CASE 9	8
CASE 10	2
CASE 11	15
CASE 12	2
CASE 13	0
CASE 14	5
CASE 15	5
CASE 16	0
CASE 17	5
CASE 18	60
CASE 19	60
CASE 20	65

In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/5.Sebastian_Rojas/Query3_regression_JSRojas.xlsx`

B.1.5 Evaluations of Judge 5

B.1.5.1 Classification

Query 1: Autism spectrum disorder in children

The similarity results of the Query 1 are presented in Table B.25. In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/1.Jhonn_Rodriguez/Query1_classification_JPRodriguez.xlsx`

Query 2: Portuguese bank telemarketing

The similarity results of the Query 2 are presented in Table B.26. In addition, the evaluation form of the Query 2 is located in:

http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/1.Jhonn_Rodriguez/Query2_classification_JPRodriguez.xlsx

Query 3: Income prediction

The similarity results of the Query 3 are presented in Table B.27. In addition, the evaluation form of the Query 3 is located in:

http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/1.Jhonn_Rodriguez/Query3_classification_JPRodriguez.xlsx

Table B.25: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 1	58
CASE 2	60
CASE 3	55
CASE 4	59
CASE 5	57
CASE 6	62
CASE 7	58
CASE 8	63
CASE 9	58
CASE 10	59
CASE 11	56
CASE 12	60
CASE 13	54
CASE 14	66
CASE 15	62
CASE 16	79
CASE 17	80
CASE 18	78
CASE 19	73
CASE 20	100
CASE 21	90
CASE 22	91
CASE 23	91
CASE 24	73
CASE 25	71

Table B.25: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 26	72
CASE 27	70
CASE 28	68
CASE 29	68
CASE 30	73
CASE 31	72
CASE 32	65
CASE 33	67
CASE 34	85
CASE 35	75
CASE 36	72

Table B.26: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 1	75
CASE 2	77
CASE 3	78
CASE 4	79
CASE 5	84
CASE 6	70
CASE 7	69
CASE 8	72
CASE 9	70
CASE 10	71
CASE 11	72
CASE 12	71
CASE 13	72
CASE 14	77
CASE 15	81
CASE 16	78
CASE 17	77
CASE 18	89
CASE 19	81
CASE 20	75
CASE 21	77
CASE 22	78
CASE 23	76

Table B.26: Query 2: Portuguese bank telemarketing

Name	Similarity (%)
CASE 24	74
CASE 25	71
CASE 26	68
CASE 27	73
CASE 28	67
CASE 29	79
CASE 30	80
CASE 31	75
CASE 32	77
CASE 33	79
CASE 34	74
CASE 35	85
CASE 36	70

Table B.27: Query 3: Income prediction

Name	Similarity (%)
CASE 1	46
CASE 2	45
CASE 3	47
CASE 4	48
CASE 5	47
CASE 6	47
CASE 7	38
CASE 8	39
CASE 9	37
CASE 10	37
CASE 11	38
CASE 12	38
CASE 13	38
CASE 14	43
CASE 15	54
CASE 16	45
CASE 17	45
CASE 18	63
CASE 19	64
CASE 20	46

Table B.27: Query 3: Income prediction

Name	Similarity (%)
CASE 21	58
CASE 22	56
CASE 23	48
CASE 24	50
CASE 25	47
CASE 26	47
CASE 27	41
CASE 28	43
CASE 29	38
CASE 30	45
CASE 31	43
CASE 32	31
CASE 33	33
CASE 34	46
CASE 35	48
CASE 36	37

B.1.5.2 Regression

Query 1: Air pollution benzene estimation
The similarity results of the Query 1 are presented in Table B.28.

Table B.28: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 1	46
CASE 2	43
CASE 3	43
CASE 4	43
CASE 5	43
CASE 6	62
CASE 7	59
CASE 8	65
CASE 9	72
CASE 10	47
CASE 11	45
CASE 12	100
CASE 13	55
CASE 14	54

Table B.28: Query 1: Air pollution benzene estimation

Name	Similarity (%)
CASE 15	48
CASE 16	45
CASE 17	59
CASE 18	58
CASE 19	59
CASE 20	58

In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/1.Jhonn_Rodriguez/Query1_regression_JPRodriguez.xlsx`

Query 2: Rental bikes hourly
The similarity results of the Query 2 are presented in Table B.29.

Table B.29: Query 2: Rental bikes hourly

Name	Similarity (%)
CASE 1	48
CASE 2	45
CASE 3	45
CASE 4	45
CASE 5	44
CASE 6	70
CASE 7	66
CASE 8	69
CASE 9	94
CASE 10	43
CASE 11	51
CASE 12	72
CASE 13	53
CASE 14	65
CASE 15	48
CASE 16	44
CASE 17	67
CASE 18	69
CASE 19	67
CASE 20	68

In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/1.Jhonn_Rodriguez/Query2_regression_JPRodriguez.xlsx`

Query 3: Coffee rust

The similarity results of the Query 3 are presented in Table B.30.

Table B.30: Query 3: Coffee rust

Name	Similarity (%)
CASE 1	38
CASE 2	37
CASE 3	37
CASE 4	38
CASE 5	38
CASE 6	53
CASE 7	50
CASE 8	61
CASE 9	61
CASE 10	47
CASE 11	53
CASE 12	52
CASE 13	56
CASE 14	46
CASE 15	51
CASE 16	36
CASE 17	58
CASE 18	55
CASE 19	58
CASE 20	55

In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/1.Jhonn_Rodriguez/Query3_regression_JPRodriguez.xlsx`

B.1.6 Evaluations of Judge 6

B.1.6.1 Classification

Query 1: Autism spectrum disorder in children

The similarity results of the Query 1 are presented in Table B.31. In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/6.Edwar_Giron/Query1_classification_EGiron.xlsx`

Query 2: Portuguese bank telemarketing

The similarity results of the Query 2 are presented in Table B.32. In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/6.Edwar_Giron/Query2_classification_EGiron.xlsx`

Query 3: Income prediction

The similarity results of the Query 3 are presented in Table B.33. In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Classification/6.Edwar_Giron/Query3_classification_EGiron.xlsx`

Table B.31: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 1	43.52
CASE 2	43.52
CASE 3	43.52
CASE 4	43.52
CASE 5	46.4
CASE 6	47.7
CASE 7	46.56
CASE 8	53.34

Table B.31: Query 1: Autism spectrum disorder in children

Name	Similarity (%)
CASE 9	49.02
CASE 10	49.14
CASE 11	46.98
CASE 12	49.32
CASE 13	45.42
CASE 14	58.32
CASE 15	47.38
CASE 16	59.1
CASE 17	59.1
CASE 18	55.82
CASE 19	53.16
CASE 20	100
CASE 21	82.14
CASE 22	84.58
CASE 23	73.38
CASE 24	62.1
CASE 25	54.9
CASE 26	61.08
CASE 27	50.3
CASE 28	55.86
CASE 29	44.6
CASE 30	50
CASE 31	50.6
CASE 32	45.08
CASE 33	49.4
CASE 34	55.06
CASE 35	56.64
CASE 36	46.4

Table B.32: Query 2: Portuguese bank telemarketing

Name	Similarity
CASE 1	41.44
CASE 2	45.7
CASE 3	48.58
CASE 4	48.58
CASE 5	45.64

Table B.32: Query 2: Portuguese bank telemarketing

Name	Similarity
CASE 6	33.68
CASE 7	35.12
CASE 8	36.32
CASE 9	35
CASE 10	33.68
CASE 11	33.68
CASE 12	35
CASE 13	35.12
CASE 14	47.6
CASE 15	52.06
CASE 16	52.28
CASE 17	52.28
CASE 18	70.36
CASE 19	55.16
CASE 20	49.84
CASE 21	47.9
CASE 22	52.54
CASE 23	39.86
CASE 24	54.76
CASE 25	49
CASE 26	41.42
CASE 27	47.88
CASE 28	39.88
CASE 29	36.58
CASE 30	43.26
CASE 31	43.26
CASE 32	41.08
CASE 33	45.4
CASE 34	46.36
CASE 35	65.5
CASE 36	34.12

Table B.33: Query 3: Income prediction

Name	Similarity
CASE 1	42.62
CASE 2	44.76

Table B.33: Query 3: Income prediction

Name	Similarity
CASE 3	45.82
CASE 4	49.76
CASE 5	48.26
CASE 6	44.36
CASE 7	44.36
CASE 8	45.56
CASE 9	45.68
CASE 10	42.92
CASE 11	44.36
CASE 12	44.24
CASE 13	44.36
CASE 14	56.84
CASE 15	62.36
CASE 16	62.96
CASE 17	61.52
CASE 18	70.92
CASE 19	66.76
CASE 20	61.78
CASE 21	68.28
CASE 22	70.58
CASE 23	61.9
CASE 24	66.5
CASE 25	59.3
CASE 26	52.52
CASE 27	55.76
CASE 28	52.84
CASE 29	36.8
CASE 30	53.94
CASE 31	53.3
CASE 32	41.68
CASE 33	44.56
CASE 34	61.12
CASE 35	63.62
CASE 36	43.36

B.1.6.2 Regression

Query 1: Air pollution benzene estimation

The similarity results of the Query 1 are presented in Table B.34.

Table B.34: Query 1: Air pollution benzene estimation

Name	Similarity
CASE 1	48.9
CASE 2	48.9
CASE 3	48.9
CASE 4	48.9
CASE 5	48.9
CASE 6	59.02
CASE 7	59.02
CASE 8	60.44
CASE 9	64.46
CASE 10	46.64
CASE 11	45.22
CASE 12	100
CASE 13	49.3
CASE 14	56.26
CASE 15	47.36
CASE 16	49.1
CASE 17	55.8
CASE 18	55.2
CASE 19	56.5
CASE 20	55.2

In addition, the evaluation form of the Query 1 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/6._Edwar_Giron/Query1_regression_EGiron.xlsx`

Query 2: Rental bikes hourly

The similarity results of the Query 2 are presented in Table B.35.

Table B.35: Query 2: Rental bikes hourly

Name	Similarity

Table B.35: Query 2: Rental bikes hourly

Name	Similarity
CASE 1	57.4
CASE 2	57.4
CASE 3	57.4
CASE 4	57.4
CASE 5	58.46
CASE 6	68.48
CASE 7	68.48
CASE 8	65.26
CASE 9	94.54
CASE 10	54.34
CASE 11	51.48
CASE 12	63.32
CASE 13	56.66
CASE 14	67.74
CASE 15	56.66
CASE 16	56.9
CASE 17	66.14
CASE 18	64.24
CASE 19	66.14
CASE 20	64.24

In addition, the evaluation form of the Query 2 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/6._Edwar_Giron/Query2_regression_EGiron.xlsx`

Query 3: Coffee rust

The similarity results of the Query 3 are presented in Table B.36.

Table B.36: Query 3: Coffee rust

Name	Similarity
CASE 1	35.7
CASE 2	35.7

Table B.36: Query 3: Coffee rust

Name	Similarity
CASE 3	35.7
CASE 4	35.7
CASE 5	35.7
CASE 6	47.42
CASE 7	47.42
CASE 8	55.42
CASE 9	48.9
CASE 10	50.78
CASE 11	39.62
CASE 12	53.8
CASE 13	51.12
CASE 14	42.88
CASE 15	50.86
CASE 16	35.9
CASE 17	47.92
CASE 18	45.2
CASE 19	47.92
CASE 20	45.2

In addition, the evaluation form of the Query 3 is located in:

`http://artemisa.unicauca.edu.co/~dcorrales/judgesPanel/Regression/6._Edwar_Giron/Query3_regression_EGiron.xlsx`

B.2 Results: Classification

Table B.37: Classification: Query 1 - Attribute-Value (Attribute).

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Autism in child	100	x	x	x	x	x	x	x
Autism in adolescent	90.941	x	x	x	x	x	x	x
Autism in adult	90.918	x	x	x	x	x	x	x
Chronic kidney disease	85.936	x	x	x	x	x	x	x
Bank telemarketing	72.061	x		x	x	x	x	x
Seismic hazard prediction	66.803	x		x	x	x	x	x
Phishing websites	62.963	x	x					x

Table B.37: Classification: Query 1 - Attribute-Value (Attribute).

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Anuran species calls	62.558	x	x					x
Anuran families calls	61.939	x	x					x
Voice rehabilitation treatment	61.867	x	x					x
Human activity recog.	61.738	x	x					x
Physical activity – 9	59.158	x	x					x
Chemi. biodegradability	57.795	x	x					
Vertebral column diagnostic	56.775	x	x					x
Physical activity – 3	56.653							x
Vertebral column injury	56.340	x	x					x
Phishing detection	55.601	x	x					x
Office occupancy	54.267	x	x					x
Physical activity – 7	53.386							x
Physical activity – 4	53.362							x
Physical activity – 6	53.129							x
Physical activity – 5	53.100							x
Physical activity – 8	53.054							x
Cardiotocography	52.911	x	x					x
Breast tissue detection	52.379	x	x					x
Default of credit card	52.229	x	x					
Physical activity – 2	51.932							x
Physical activity – 1	51.653							x
Ozone level 8 hours	47.601	x						x
Ozone level 1 hour	40.420	x						x
Companies bankruptcy 5	39.589	x						
Companies bankruptcy 4	37.248	x						x
Companies bankruptcy 3	36.192	x						x
Companies bankruptcy 2	34.727	x						x
Companies bankruptcy 1	34.669	x						x
Risk factors cervical cancer	32.813	x						x

Table B.38: Classification: Query 1 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Autism in child	100	x	x	x	x	x	x	x
Autism in adolescent	88.018	x	x	x	x	x	x	x
Autism in adult	87.670	x	x	x	x	x	x	x

Table B.38: Classification: Query 1 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Chronic kidney disease	71.909	x	x	x	x	x	x	x
Bank telemarketing	68.010	x		x	x	x	x	x
Seismic hazard prediction	63.531	x		x	x	x	x	x
Anuran species calls	60.093	x	x					x
Anuran families calls	59.629	x	x					x
Phishing websites	57.853	x	x					x
Voice rehabilitation treatment	56.919	x	x					x
Physical activity – 9	56.728	x	x					x
Chemi. biodegradability	55.924	x	x					
Physical activity – 3	55.530							x
Vertebral column diagnostic	54.907	x	x					x
Human activity recog.	54.154	x	x					x
Vertebral column injury	53.587	x	x					x
Physical activity – 7	53.021							x
Physical activity – 4	52.870							x
Physical activity – 5	52.853							x
Physical activity – 6	52.768							x
Physical activity – 8	52.717							x
Phishing detection	52.494	x	x					x
Cardiotocography	52.304	x	x					x
Office occupancy	52.046	x	x					x
Physical activity – 2	51.940							x
Physical activity – 1	51.789							x
Default of credit card	49.214	x	x					
Ozone level 8 hours	48.888	x						x
Breast tissue detection	47.982	x	x					x
Ozone level 1 hour	43.502	x						x
Companies bankruptcy 5	39.391	x						
Companies bankruptcy 4	37.365	x						x
Companies bankruptcy 3	36.520	x						x
Risk factors cervical cancer	36.326	x						x
Companies bankruptcy 1	35.609	x						x
Companies bankruptcy 2	35.395	x						x

Table B.39: Classification: Query 2 - Attribute-Value (Attribute)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Seismic hazard prediction	95.129	x	x	x	x	x	x	x
Bank telemarketing	94.755	x	x	x	x	x	x	x
Chronic kidney disease	75.871	x		x	x	x	x	x
Autism in adolescent	72.226	x		x	x	x	x	x
Autism in adult	67.626	x		x	x	x	x	
Autism in child	62.362	x		x	x	x	x	x
Risk factors cervical cancer	60.263	x	x					x
Companies bankruptcy 5	51.838	x	x					x
Cardiotocography	50.01	x						x
Ozone level 8 hours	49.697		x					
Companies bankruptcy 4	49.402		x					x
Chemi. biodegradability	48.518	x						x
Companies bankruptcy 3	47.663	x	x					x
Default of credit card	47.158	x						x
Voice rehabilitation treatment	46.975	x						x
Companies bankruptcy 2	46.346	x	x					
Companies bankruptcy 1	45.965	x	x					x
Physical activity – 9	45.098	x						x
Ozone level 1 hour	42.511	x	x					
Phishing websites	42.507	x						x
Human activity recog.	41.407	x						x
Physical activity – 1	39.696							x
Phishing detection	39.582	x						
Office occupancy	39.028	x						x
Anuran families calls	38.189	x						
Anuran species calls	37.568	x						x
Vertebral column diagnostic	37.324	x						x
Breast tissue detection	36.921	x						x
Physical activity – 3	36.249							x
Physical activity – 2	35.843							x
Physical activity – 7	35.376							x
Vertebral column injury	35.263	x						x
Physical activity – 8	35.208							x
Physical activity – 6	35.106							x
Physical activity – 5	33.654							x
Physical activity – 4	33.066							x

Table B.40: Classification: Query 2 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Bank telemarketing	81.056	x	x	x	x	x	x	x
Seismic hazard prediction	79.515	x	x	x	x	x	x	x
Autism in adolescent	55.038	x		x	x	x	x	x
Autism in adult	51.428	x		x	x	x	x	
Chronic kidney disease	49.066	x		x	x	x	x	x
Autism in child	48.948	x		x	x	x	x	x
Risk factors cervical cancer	41.536	x	x					x
Physical activity – 9	37.842	x						x
Companies bankruptcy 5	35.73	x	x					x
Physical activity – 1	34.631							x
Companies bankruptcy 4	33.998		x					x
Default of credit card	33.044	x						x
Companies bankruptcy 3	32.721	x	x					x
Physical activity – 3	32.498							x
Ozone level 8 hours	32.029		x					
Chemi. biodegradability	32.006	x						x
Companies bankruptcy 2	31.763	x	x					
Physical activity – 7	31.627							x
Physical activity – 2	31.598							x
Cardiotocography	31.507	x						x
Companies bankruptcy 1	31.352	x	x					x
Physical activity – 6	31.333							x
Physical activity – 8	31.274							x
Voice rehabilitation treatment	30.719	x						x
Physical activity – 5	30.114							x
Physical activity – 4	29.757							x
Ozone level 1 hour	26.639	x	x					
Office occupancy	26.056	x						x
Phishing websites	25.614	x						x
Anuran families calls	23.985	x						
Anuran species calls	23.519	x						x
Vertebral column diagnostic	23.254	x						x
Human activity recog.	22.216	x						x
Phishing detection	21.915	x						
Vertebral column injury	20.714	x						x
Breast tissue detection	19.271	x						x

Table B.41: Classification: Query 3 - Attribute-Value (Attribute)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Bank telemarketing	84.913	x		x	x	x	x	x
Chronic kidney disease	84.820	x	x	x	x	x	x	x
Autism in adolescent	84.559	x	x	x	x	x	x	x
Autism in adult	83.268	x	x	x	x	x	x	x
Seismic hazard prediction	83.066	x		x	x	x	x	
Autism in child	76.977		x	x	x	x	x	x
Chemi. biodegradability	60.872	x	x					x
Voice rehabilitation treatment	59.405	x	x					x
Physical activity – 9	58.055	x	x					x
Default of credit card	56.877	x	x					
Phishing websites	56.400	x	x					x
Cardiotocography	56.243	x	x					x
Anuran families calls	55.138	x	x					x
Office occupancy	54.806	x	x					x
Anuran species calls	54.517	x	x					x
Human activity recog.	52.721	x	x					x
Vertebral column diagnostic	52.568	x	x					x
Phishing detection	52.272	x	x					x
Risk factors cervical cancer	51.816	x						
Vertebral column injury	50.507	x	x					x
Physical activity – 1	49.759							x
Physical activity – 3	49.444							
Physical activity – 2	49.253							x
Physical activity – 6	48.661							x
Breast tissue detection	48.244	x	x					x
Physical activity – 8	48.135							x
Physical activity – 4	47.677							x
Physical activity – 7	47.664							x
Physical activity – 5	47.178							x
Ozone level 8 hours	45.519	x						
Companies bankruptcy 5	40.591	x						x
Ozone level 1 hour	38.333	x						
Companies bankruptcy 4	38.119	x						x
Companies bankruptcy 3	36.915	x						x
Companies bankruptcy 2	35.887	x						
Companies bankruptcy 1	35.504	x						x

Table B.42: Classification: Query 3 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Bank telemarketing	81.789	x		x	x	x	x	x
Autism in adolescent	80.498	x	x	x	x	x	x	x
Autism in adult	78.788	x	x	x	x	x	x	x
Seismic hazard prediction	77.975	x		x	x	x	x	
Autism in child	72.833		x	x	x	x	x	x
Chronic kidney disease	72.317	x	x	x	x	x	x	x
Chemi. biodegradability	57.36	x	x					x
Default of credit card	56.347	x	x					
Voice rehabilitation treatment	55.516	x	x					x
Office occupancy	54.113	x	x					x
Physical activity – 9	53.725	x	x					x
Anuran families calls	51.892	x	x					x
Phishing websites	51.632	x	x					x
Anuran species calls	51.426	x	x					x
Cardiotocography	51.257	x	x					x
Vertebral column diagnostic	50.735	x	x					x
Risk factors cervical cancer	50.214	x						
Physical activity – 3	48.341							
Vertebral column injury	48.194	x	x					x
Physical activity – 6	47.713							x
Physical activity – 2	47.707							x
Physical activity – 8	47.171							x
Physical activity – 7	46.878							x
Physical activity – 4	46.518							x
Phishing detection	46.455	x	x					x
Physical activity – 1	46.348							x
Physical activity – 5	46.232							x
Human activity recog.	46.099	x	x					x
Breast tissue detection	44.41	x	x					x
Ozone level 8 hours	43.666	x						
Companies bankruptcy 5	41.352	x						x
Companies bankruptcy 4	39.499	x						x
Companies bankruptcy 3	38.479	x						x
Ozone level 1 hour	38.276	x						
Companies bankruptcy 1	37.596	x						x
Companies bankruptcy 2	37.476	x						

B.3 Results: Regression

Table B.43: Regression: Query 1 - Attribute-Value (Attribute)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Air pollution – benzene estimation	100	x	x	x	x	x	x	x
Airfoil self–noise	80.543	x	x	x				x
II-Room temperature	78.419	x	x	x	x	x	x	x
Rental bikes hourly	78.297	x	x	x	x	x		x
II-Dinning room temperature	78.261	x	x	x	x	x	x	x
I-Dinning room temperature	77.589	x	x	x	x	x		x
I-Room temperature	77.57	x	x	x	x	x		x
Compressor decay	75.412	x	x	x	x	x		x
Turbine decay	75.356	x	x	x	x	x		x
Rental bikes daily	74.963	x	x	x	x	x	x	x
Feedback blogs prediction	73.854							x
Comments prediction in FB – 1	72.41							x
Comments prediction in FB – 2	71.862							x
Comments prediction in FB – 5	71.828							x
Comments prediction in FB – 3	71.572							
Comments prediction in FB – 4	71.27							x
Energy use of appliances	47.505	x	x					x
Posts in Facebook pages	45.678	x	x					x
Predict the forest fires	39.636							x
Beijing PM 2.5	35.946	x	x					x

Table B.44: Regression: Query 1 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Air pollution – benzene estimation	100	x	x	x	x	x	x	x
Airfoil self–noise	81.445	x	x	x				x
II-Room temperature	78.481	x	x	x	x	x	x	x
II-Dinning room temperature	78.443	x	x	x	x	x	x	x
I-Dinning room temperature	78.413	x	x	x	x	x		x
I-Room temperature	78.343	x	x	x	x	x		x
Compressor decay	77.754	x	x	x	x	x		x
Turbine decay	77.753	x	x	x	x	x		x
Feedback blogs prediction	77.409							x

Table B.44: Regression: Query 1 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Rental bikes hourly	75.649	x	x	x	x	x		x
Comments prediction in FB – 5	75.537							x
Comments prediction in FB – 2	74.957							x
Comments prediction in FB – 1	74.922							x
Comments prediction in FB – 4	74.777							x
Comments prediction in FB – 3	74.133							
Rental bikes daily	72.522	x	x	x	x	x	x	x
Energy use of appliances	49.663	x	x					x
Posts in Facebook pages	42.463	x	x					x
Predict the forest fires	42.317							x
Beijing PM 2.5	38.888	x	x					x

Table B.45: Regression: Query 2 - Attribute-Value (Attribute)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Rental bikes hourly	97.452	x	x	x	x	x	x	x
Airfoil self–noise	87.263	x	x	x				x
I-Dinning room temperature	85.871	x	x	x	x	x	x	x
I-Room temperature	85.854	x	x	x	x	x	x	
II-Dinning room temperature	85.306	x	x	x	x	x		
II-Room temperature	85.299	x	x	x	x	x		x
Rental bikes daily	84.148	x	x	x	x	x		x
Compressor decay	81.074	x	x	x	x	x	x	x
Turbine decay	81.012	x	x	x	x	x	x	x
Feedback blogs prediction	76.806							x
Air pollution – benzene estimation	76.323	x	x	x	x	x		x
Comments prediction in FB – 1	76.017							x
Comments prediction in FB – 2	75.362							x
Comments prediction in FB – 3	75.187							x
Comments prediction in FB – 5	75.167							x
Comments prediction in FB – 4	74.892							
Beijing PM 2.5	49.333	x	x					x
Posts in Facebook pages	48.746	x	x					
Energy use of appliances	48.095	x	x					x
Predict the forest fires	41.722							x

Table B.46: Regression: Query 2 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Rental bikes hourly	96.913	x	x	x	x	x	x	x
Airfoil self–noise	86.396	x	x	x				x
I-Dinning room temperature	85.34	x	x	x	x	x	x	x
I-Room temperature	85.266	x	x	x	x	x	x	
II-Dinning room temperature	84.836	x	x	x	x	x		
Rental bikes daily	84.37	x	x	x	x	x		x
II-Room temperature	83.917	x	x	x	x	x		x
Compressor decay	81.724	x	x	x	x	x	x	x
Turbine decay	81.719	x	x	x	x	x	x	x
Feedback blogs prediction	77.869							x
Comments prediction in FB – 5	75.542							x
Comments prediction in FB – 3	75.425							x
Comments prediction in FB – 4	75.295							
Comments prediction in FB – 1	75.203							x
Comments prediction in FB – 2	75.177							x
Air pollution – benzene estimation	73.624	x	x	x	x	x		x
Beijing PM 2.5	55.003	x	x					x
Energy use of appliances	50.187	x	x					x
Posts in Facebook pages	47.984	x	x					
Predict the forest fires	43.49							x

Table B.47: Regression: Query 3 - Attribute-Value (Attribute)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Energy use of appliances	87.94	x	x	x	x			x
Beijing PM 2.5	85.733	x	x	x	x			x
Posts in Facebook pages	80.502	x	x	x	x			x
Predict the forest fires	76.564			x	x			x
Rental bikes hourly	54.588	x	x			x		x
Airfoil self–noise	51.967	x	x					x
II-Dinning room temperature	47.403	x	x			x	x	x
II-Room temperature	47.383	x	x			x	x	x
Air pollution – benzene estimation	47.369	x	x			x	x	x
I-Dinning room temperature	46.638	x	x			x		x
I-Room temperature	46.572	x	x			x		x
Feedback blogs prediction	45.668							x

Table B.47: Regression: Query 3 - Attribute-Value (Attribute)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Turbine decay	45.461	x	x			x		x
Compressor decay	45.136	x	x			x		x
Rental bikes daily	44.24	x	x			x	x	x
Comments prediction in FB – 5	41.687							x
Comments prediction in FB – 2	41.641							x
Comments prediction in FB – 3	41.536							x
Comments prediction in FB – 4	41.473							x
Comments prediction in FB – 1	41.42							x

Table B.48: Regression: Query 3 - Attribute-Value (Dataset)

Name	Similarity	Clusters - K						Quartile
		2	3	4	5	6	7	
Energy use of appliances	85.704	x	x	x	x			x
Beijing PM 2.5	82.258	x	x	x	x			x
Predict the forest fires	76.295			x	x			x
Posts in Facebook pages	73.794	x	x	x	x			x
Rental bikes hourly	52.064	x	x			x		x
Airfoil self–noise	48.8	x	x					x
II-Dinning room temperature	46.911	x	x			x	x	x
II-Room temperature	46.869	x	x			x	x	x
Air pollution – benzene estimation	45.868	x	x			x	x	x
I-Dinning room temperature	45.363	x	x			x		x
I-Room temperature	45.31	x	x			x		x
Rental bikes daily	45.104	x	x			x	x	x
Turbine decay	44.487	x	x			x		x
Compressor decay	44.25	x	x			x		x
Feedback blogs prediction	42.925							x
Comments prediction in FB – 5	38.941							x
Comments prediction in FB – 2	38.839							x
Comments prediction in FB – 3	38.718							x
Comments prediction in FB – 4	38.714							x
Comments prediction in FB – 1	38.626							x

C. Prototype: Hygeia data

Chapters 3, 4 and 5 described the Framework for Data Quality in Knowledge Discovery Tasks (classification and regression). In this chapter we explain the prototype called Hygeia data, which implements the proposed approaches. The tool guides to the user in the data cleaning process, also Hygeia recommends the suitable data cleaning algorithms respect a user dataset.

C.1 System Functionalities

Given a new dataset of a user, the goal of Hygeia data tool is to recommend the suitable data cleaning algorithms. The system is presented in Figure C.1 . This is composed of the following modules:

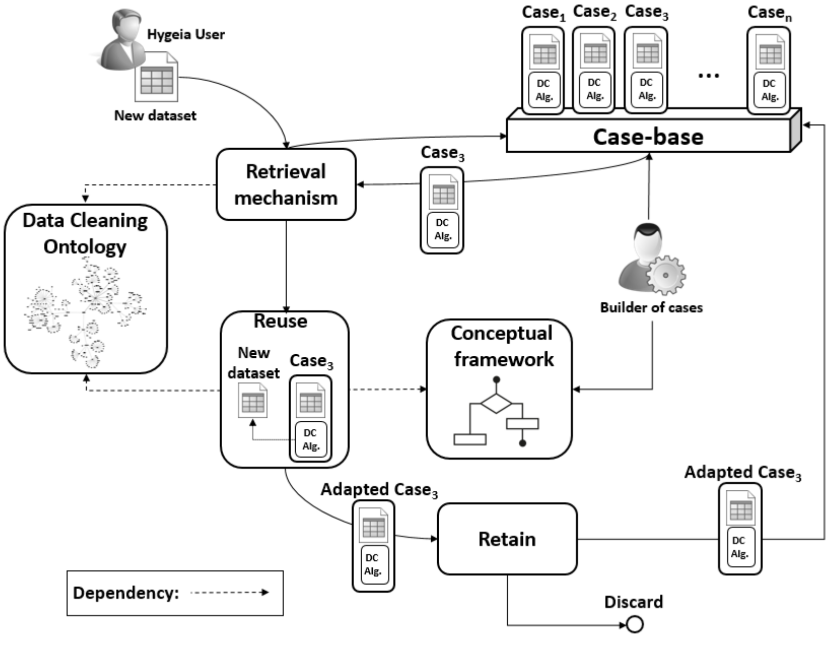


Figure C.1: Hygeia data tool

- **Case-base** contains a set of cases. A case is represented by a dataset and the algorithms (DC Alg.) used to clean it. Figure C.1 the case–base contains the cases: $Case_1$, $Case_2$, $Case_3$, ... , $Case_n$.
- **Retrieval mechanism** compares the new dataset against the datasets of the case-base, and this selects the most similar dataset of the case-base respect to the new dataset. For example, in Figure C.1 the retrieval mechanism selects the $Case_3$.
- **Reuse module**, the data cleaning algorithms of the selected dataset are used to clean the new dataset. Thus, a new case is created. In Figure C.1, the data cleaning algorithms (DC Alg.) of $Case_3$ are used in the new dataset, then the new case is named $AdaptedCase_3$.
- **Data cleaning ontology** plays a key role in the Hygeia data tool. This supports the reuse module in the recommendation of similar data cleaning algorithms to the used in the dataset of the $Case_3$.
- **Retain module**, the new case $AdaptedCase_3$ is stored in the case–base, if data cleaning algorithms used in the new dataset obtained a good performance, in otherwise $AdaptedCase_3$ is discarded.
- **Conceptual framework** is used for building the cases of the case-base, also the conceptual framework guides to the Hygeia user in the data cleaning process based on the solution of the retrieved $Case_3$.

C.2 System Architecture

The system architecture of Hygeia data tool is represented by a logical view shown in Figure C.2. This view organizes the software classes into packages and three layers: Application, Mediation and Foundation [256, 257]. Figure C.2 depicts the layers of the Hygeia architecture and the interaction among packages.

C.2.1 Application layer

The Application layer provides the functionalities to a Hygeia user. This layer is composed by the package:

- **Graphical user interface** which contains the software classes and forms to achieve a visual representation. This enables a user interacts with the Hygeia tool functionalities through graphical elements, such as text, windows, icons, buttons, text fields, combo box etc. We developed the forms with Swing API in NetBeans IDE 8.2 [3].

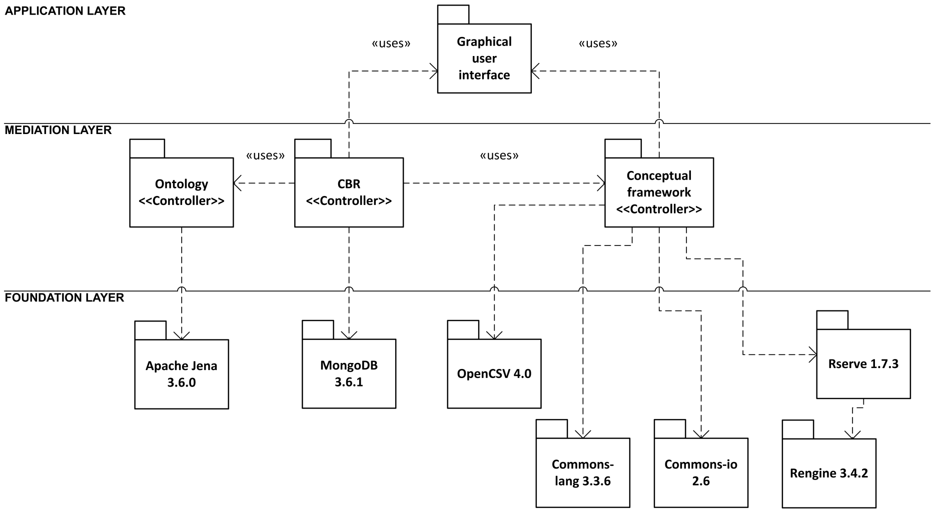


Figure C.2: Logical view of Hygeia data tool

C.2.2 Mediation layer

The mediation layer contains software classes named controllers. These represent the logical of Ontology, CBR and Conceptual Framework, also the controllers take user requests and pass it into the foundation layer.

- **Ontology** contains a set of software classes which mapping the structure of the ontology. The mapped classes allow the communication between data cleaning ontology and CBR controllers.

- **CBR** implements the Retrieval mechanism, Reuse and Retain modules through software classes. Additionally, this package sends to Graphical User Interface the retrieved case of the case–base.

- **Conceptual framework** is composed by a set of software classes for guiding the user in the data cleaning process, also this package request the parameters of data cleaning methods from Graphical User Interface and it sends the result of data cleaning methods to Graphical User Interface.

C.2.3 Foundation layer

The foundation layer is represented by the software used in the Hygeia data tool.

- **Apache Jena 3.6.0** is a Java framework. This includes software functionalities for RDF, RDFS, OWL, SPARQL, also an inference engine [258]. Apache Jena allows the communication between Data cleaning ontology and the Ontology Controllers.

- **MongoDB 3.6.1** is a NoSQL database. This stores data in JSON documents [259]. We used MongoDB as backup of the case-base, also the discarded cases are stored in the mongoDB. The case-base is located in: `http://artemisa.unicauca.edu.co/~dcorrales/case-base/cb_v.0.6.tar`.

- **OpenCSV 4.0** is a CVS parser library for Java [260]. It was used for preprocessing of the new datasets in Conceptual Framework.

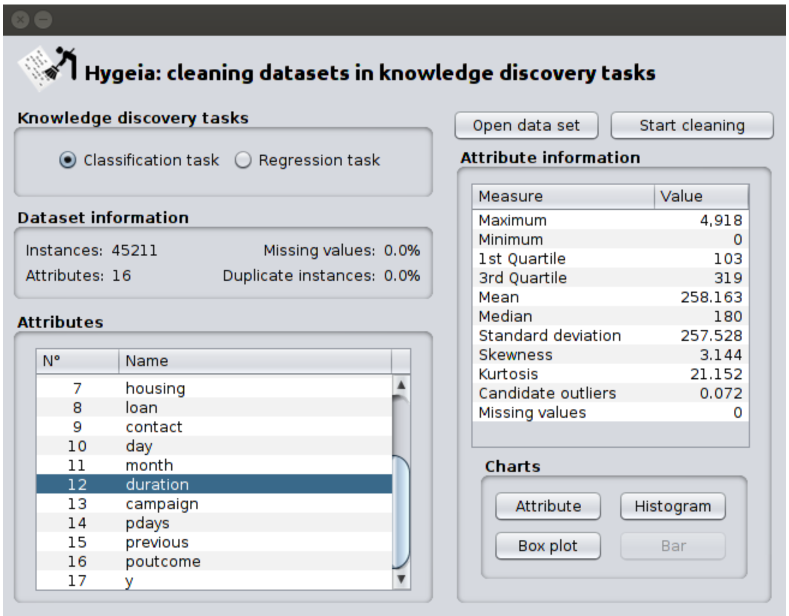
- **Commons–lang 3.3.6 and Commons–io 2.6** provide utilities in Java, directly in String manipulation, numerical methods, creation and serialization and System properties [261].

- **Rserve 1.7.3** Rserve acts as a socket server (TCP/IP or local sockets) which responds to requests from Conceptual Framework controllers. It listens for any incoming connections and processes incoming requests [262]. In other words, Rserve allows to embed R code within Conceptual Framework controllers.

- **Rengine** is an engine of R statistical program [188]. The data cleaning algorithms and charts belong to R packages, they are collections of functions developed by the R community. We used R version 3.4.2 with missForest and mice packages [263, 264] for imputation task, Rlof [265] and fpc [266] packages for outliers detection task, UBL and smotefamily packages [267, 268] for balanced classes and Fselector [269] package for dimensionality reduction tasks. In case of remove duplicate instances and label correction, we used R primary functions.

C.3 User Interfaces

In this section the Graphical User Interfaces are presented. We developed two main forms. The first form presents statistic information related with the dataset (number of attributes and instances,percentage of missing values and duplicate instances) and its attributes (mean, median, skewness, kurtosis, etc.) as show Figure C.3. In addition, this form offers charts for attributes as Histogram, Box plot, Bars, and Line (Figure C.5).



Hygeia: cleaning datasets in knowledge discovery tasks

Knowledge discovery tasks

● Classification task ○ Regression task

Open data set | Start cleaning

Dataset information

Instances: 45211 Missing values: 0.0%

Attributes: 16 Duplicate instances: 0.0%

Attributes

N°	Name
7	housing
8	loan
9	contact
10	day
11	month
12	duration
13	campaign
14	pdays
15	previous
16	poutcome
17	y

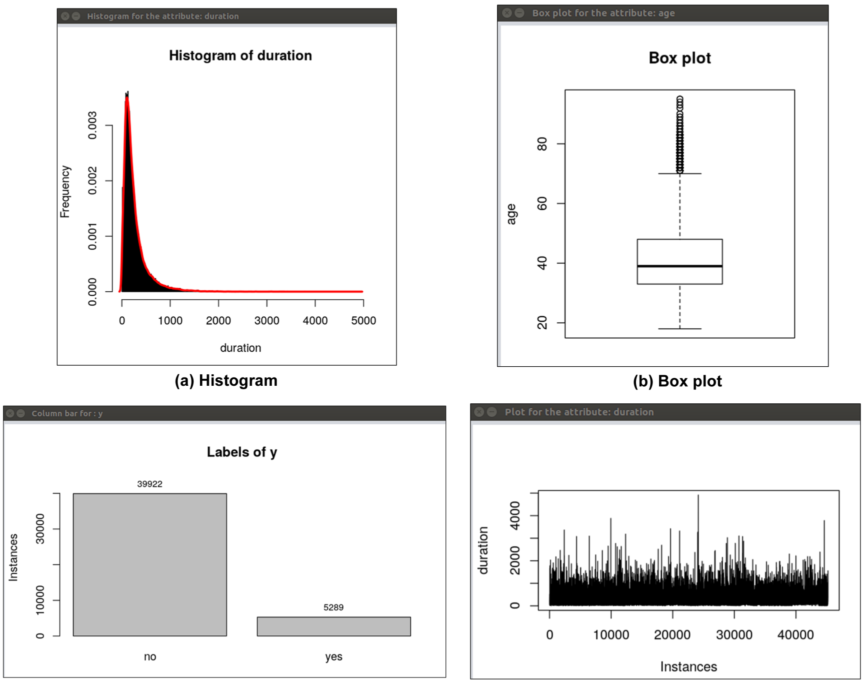
Attribute information

Measure	Value
Maximum	4,918
Minimum	0
1st Quartile	103
3rd Quartile	319
Mean	258.163
Median	180
Standard deviation	257.528
Skewness	3.144
Kurtosis	21.152
Candidate outliers	0.072
Missing values	0

Charts

Attribute | Histogram | Box plot | Bar

Figure C.3: Form of the statistical information of a dataset.



(a) Histogram

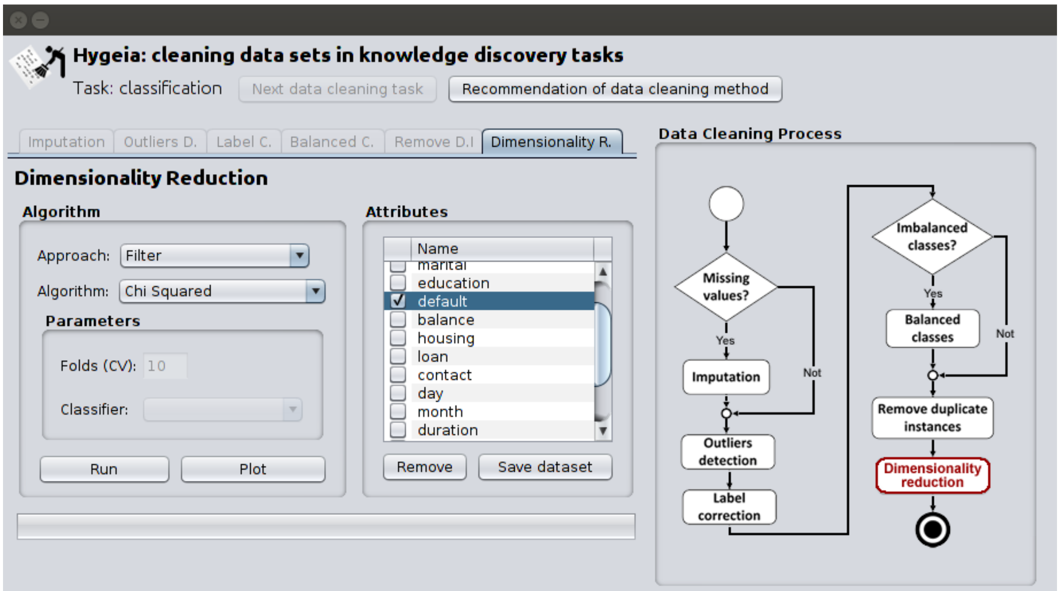
(b) Box plot

(c) Bars

(d) Line

Figure C.4: Charts for attributes: Histogram, Box plot, Bars

The second form corresponds to conceptual framework for classification and regression tasks. The conceptual framework form appears when the "Start cleaning" button is pressed and the radio button of knowledge discovery tasks is selected (Form depicted in Figure C.3). Figure C.5 shows the conceptual framework for classification tasks when the chi–squared algorithm is applied in the dimensionality reduction phase. The "Plot" button depicts the results of chi-squared algorithm as show Figure C.6b.



Hygeia: cleaning data sets in knowledge discovery tasks

Task: classification

Dimensionality Reduction

Algorithm

Approach: Filter

Algorithm: Chi Squared

Parameters

Folds (CV): 10

Classifier:

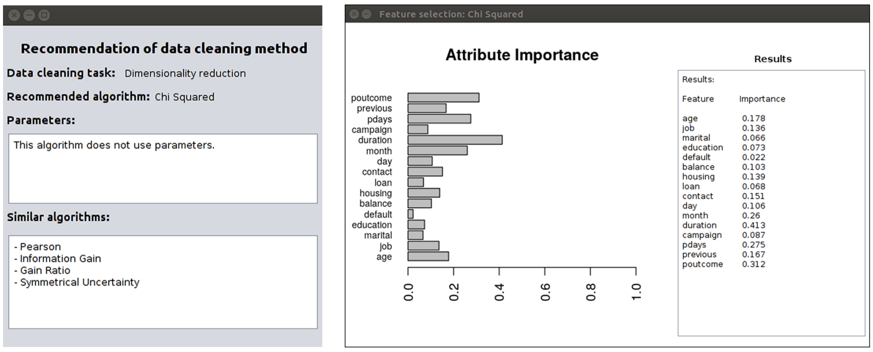
Attributes

Name

- □ marital
- □ education
- ☑ default
- □ balance
- □ housing
- □ loan
- □ contact
- □ day
- □ month
- □ duration

Figure C.5: Conceptual framework form

Additionally, the conceptual framework form (Figure C.5) contains a "Recommendation of data cleaning" button which represents the case–base reasoning system. Figure C.6a presents the data cleaning algorithm of the retrieved case, and similar data cleaning algorithms inferred by Data cleaning ontology.



Recommendation of data cleaning method

Data cleaning task: Dimensionality reduction

Recommended algorithm: Chi Squared

Parameters:

This algorithm does not use parameters.

Similar algorithms:

- Pearson
- Information Gain
- Gain Ratio
- Symmetrical Uncertainty

Feature selection: Chi Squared

Results

```
Results:

Feature       Importance

age            0.178
job            0.136
marital        0.066
education      0.073
default        0.022
balance        0.103
housing        0.139
loan           0.068
contact        0.151
day            0.106
month          0.26
duration       0.413
campaign       0.087
pdays          0.275
previous       0.167
poutcome       0.312
```

(a) Recommendation of data cleaning method

(b) Results of Chi squared method

Figure C.6: Forms of recommendation of data cleaning method and results of chi–squared.