Volume 6, Issue 10, October 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Working with Data: Important Cleaning Procedures

[1]George Anderson*, [2]Audrey Masizana, [3]Dimane Mpoeleng
[1, 2] Department of Computer Science, University of Botswana, Botswana
[3] Department of Computer Science and Information Systems, Botswana International University of Science and Technology, Botswana

Abstract— Generally speaking, "statistics is concerned with the transformation of raw data into knowledge". Statistics uses uncertainty techniques to decide what data should be collected, how much data should be collected, to draw conclusions, to determine how far such conclusions can be trusted. Data matching (record linkage) and deduplication, is concerned with taking two lists of records, representing the same objects, and matching them. We consider the case of students writing exams in a university using Optical Mark Recognition forms. Their exam records can contain errors, due to incorrect completion; however, these records have to be matching with course registration lists to make sure that the right student is awarded the right mark. The process intends to identify and correct these errors or at least to minimize their impact on the results. In our prior work, we have come up with a methodology to solve this problem, based on reducing the length of the two lists of records, before applying a process of blocking (grouping student records), comparison (comparing record pairs), and matching (identifying matching pairs). Our second contribution is a summary of application of missing data imputation, especially when data is sourced integrated from multiple sources in a data warehouse. Business evaluations and other decisions tend to break down when data is missing. Ignoring this can lead to biased results in statistical modeling. Methods have been developed to estimate missing data so that the data can be used for tasks such as pattern recognition or business decision making. As a third contribution, we consider missing data imputation and data matching, and propose a novel framework which integrates the two data cleaning approaches. Without missing data imputation and data matching, many data sets become unusable.

Keywords— Data Cleaning, Preprocessing for Statistics, Data Matching, Missing Data Imputation.

I. INTRODUCTION

Generally speaking, "statistics is concerned with the transformation of raw data into knowledge" [1]. When faced with the task of analysing raw data, statistics helps decide on what data should be collected, how much data should be collected, what conclusions one can arrive at by examining the data and how far those conclusions can be relied on. Statistics makes use of uncertainty techniques to address these issues. Many statistical techniques such as regression and hypothesis testing are well known to different kinds of scientists and are used in numerous fields, such as Computer Science, Biology, Electrical Engineering, Social Work, Educational Foundations, Chemistry, Business, Information Systems, etc. Statistics fits into a general process of investigations [2]:

i. Identify a problem, such as predicting prices of houses in a city.
ii. Collect relevant data on the topic, such as past prices of houses and their specifications.
iii. Analyze the data, using Linear Regression, for example.
iv. Arrive at a conclusion, for example, a model, with evaluation metrics.

Once stage (ii) is complete, there is a need to prepare the data for analysis. Data analysis could be done with statistical software such as R [3], for example. In this paper, we focus on the process of preparing data for analysis known as Data Cleaning. Data cleaning, data cleansing, and data scrubbing refers to the process of detecting and removing errors and inconsistencies from data in order to improve the quality of data [4]. Data quality problems exist in single data collections, such as databases. These could include misspelled, missing, erroneous, duplicated data, etc. When multiple data sources, have to be integrated, the cleaning problem becomes much more complex, because integration involves data extracted from disparate systems with heterogeneous data sources having their own distinct set of characteristics and representations.

One of the techniques used in data cleaning is data matching also known as record linkage. Data matching solves the problem of identifying and matching records from disparate databases that refer to the same real-world entities [5]. Applications include E-Commerce comparative shopping, where the same item which appears in several online stores might have slightly different values for attributes such as descriptions; there is a need to match the items so it is able to compare prices. Patient records in different lists, for different hospitals for example, may also have errors in names or identification numbers. By matching records in different lists correctly, tasks such as identifying adverse drug reactions are enhanced.

One focus area of data cleaning is the application of a statistical method called missing data imputation, where missing values are handled by filling-in (imputing) one or more plausible values. It deals with the problem of arriving at values for record fields that have no data [6]. Certain statistical algorithms, such as linear regression, cannot work if data is missing in fields in entity records.

In this paper we wish to explore how the data matching technique could merge with missing data imputation to provide an effective data cleaning technique. Our contributions are as follows: i) We summarize data matching; and ii) missing data imputation approaches in the context of data cleaning; iii) We show how these the processes in these two fields could be of benefit to one another by proposing a framework which integrates the two processes. To the best of our knowledge, this has not been done before.

The rest of this paper is organized as follows: Section II provides more background on the general area of Data Cleaning; Section III discusses Data Matching, including some results obtained in prior work done by the authors; Section IV discusses Missing Data Imputation; Section V proposes a framework which unifies Data Matching and Missing Data Imputation and proposes ways in which each area can be of benefit to the other; Section VI has our conclusion.

II. BACKGROUND ON DATA CLEANING

The amount of data that exists in the world doubles roughly every 18 months [7]. Big Data issues come up at the high end of the scale, with huge volumes of data that have to be processed and stored. At the other end, individual sources of data are maintained in small collections, such as in a single spreadsheet. Regardless of size, if data is to be used, whether directly by humans or processed automatically by machines, there is a need ensure the data does not have errors in it, or other problems which might affect use adversely [8]-[10]. With Big Data, there as an even greater need for clean data, since erroneous data could be very costly. The cleaning process determines inaccurate or incomplete data and then improves the quality through correcting of the detected errors and omissions. In this section, we discuss two issues pertaining to data cleaning; data warehousing and the use of statistical processes.

A. Data Cleaning Applications in Databases

A typical application of data cleaning in the field of database technology is experienced in Data Warehouse development. A data warehouse is a database takes data from multiple heterogeneous sources and integrates it together to present a global enterprise view. The process involves extracting data from multiple heterogeneous sources, cleaning the data by finding errors and correcting the errors in the data and finally converting the data into the format that support analytical reporting structured for decision making. Data warehousing assists knowledge workers, such as managers and analysts, to make good and fast decisions [11].

Data warehousing is used in various industries, such as manufacturing (for order shipment), retail (for user profiling), financial services (for risk analysis and credit card fraud detection) and healthcare (for outcome analysis). Data warehousing supports OLAP (Online Analytical Processing) in which summarized data as well as data consolidated from multiple databases is used to support decision making. In order for appropriate decisions to be made, it is very important that the data in a data warehouse be correct, hence the data cleaning stage of data warehouse development process is crucial.

Since data warehousing involves high volumes of data coming from multiple sources, there is a high likelihood of errors. Data cleaning must fix the errors in all data sources as when integrating multiple sources [4]. Therefore data cleaning can have a huge payoff. Examples of errors in such environments include inconsistent value assignments, missing entries, and inconsistent descriptions. Some data warehouse data cleaning tools make use of domain knowledge (e.g. for postal addresses) and some sort of approximate matching to correct data being imported from multiple sources. Other tools make use of data mining technology to report suspicious patterns, indicating incorrect data (e.g. that a certain retail store has never received any complaints).

The approach must be supported by tools to minimize manual work and time consuming programming, and must be extensible such that it can be used for other sources. For large processes such as data warehousing, data cleaning tools must support the entire workflow infrastructure. Table I shows examples of dirty data problems at schema level i.e. those which violate integrity constraints which are specified in a database to ensure data is correct. Table II shows examples of dirty problems at instance level, which cannot be detected via checking of integrity constraints.

Table I Examples of Dirty Data at Schema Level [4]

Scope/Problem		Dirty Data	Reasons/Remarks
Attribute	Illegal values	bdate=30.13.70	values outside of domain range
Record	Violated attribute dependencies	age=22, bdate=12.02.70	age = current year - birth year should hold
Record type	Uniqueness violation	emp_1=(name="John Smith", SSN="123456"); emp_2=(name="Peter Miller", SSN="123456")	uniqueness for SSN (social security number) violated
Source	Referential integrity violation	emp=(name="John Smith", deptno=127)	referenced department (127) not defined

Table II Examples of Dirty Data at Instance Level [4]

Scope/Problem		Dirty Data	Reasons/Remarks
Attribute	Missing values	phone=9999-999999	unavailable values during data entry (dummy values or null)
	Misspellings	city="Liipzig"	usually typos, phonetic errors
	Cryptic values, Abbreviations	experience="B"; occupation="DB Prog."	
	Embedded values	name="J. Smith 12.02.70 New York"	multiple values entered in one attribute (e.g. in a free-form field)
	Misfielded values	city="Germany"	
Record	Violated attribute dependencies	city="Redmond", zip=77777	city and zip code should correspond
Record type	Word transpositions	$name_1$= "J. Smith", $name_2$="Miller P."	usually in a free-form field
	Duplicated records	emp_1=(name="John Smith",...); emp_2=(name="J. Smith",...)	same employee represented twice due to some data entry errors
	Contradicting records	emp_1=(name="John Smith", bdate=12.02.70); emp_2=(name="John Smith", bdate=12.12.70)	the same real world entity is described by different values
Source	Wrong references	emp=(name="John Smith", deptno=17)	referenced department (17) is defined but wrong

B. Data Cleaning Approaches and Applications in Statistics

Many statistical analyses try to find a pattern in a data series based on a hypothesis or assumption about the data. Cleaning could involve removing data which is either i) disconnected from the effect or assumption we are trying to isolate, due to some factors relevant to only that data; or ii) obviously erroneous e.g. due to a mistake in data collection [12]. Data points to be cleaned are usually outliers i.e. do not conform to a generally visible pattern. They could be identified via a plot, for example, since they might lie far outside of the general distribution. They could also be identified by running the analysis on the entire data set and eliminating points which do not meet mathematical limits for variability from a trend, and then repeating the analysis on the remaining data. Cleaning could be done manually by identifying data coming from a problematic source e.g. from a business unit which has misreported sales figures in the past. By some accounts, good statisticians need to spend 90% of their time collecting and cleaning data and developing hypothesis, and 10% on actual mathematical manipulation of the data. By some other accounts 50% of the time needed for analysis is spent on data cleaning.

Table III shows data for ACPRVF/M (percent females/males with low arm circumference), by district, for some country. One can see that the maximum values are quite high. Figure 1 shows ACPRVM data in the form of a histogram, with the erroneous data visible as an outlier. Figure 2 shows a scatterplot of height vs age data, clearly showing outliers. With the help of tools such as histograms and scatterplots, erroneous data points which manifest themselves as outliers can be identified, and prevented from causing mistakes in analysis. Once the data is cleaned, statistical methods such as histogram analysis, scatterplot analysis, linear regression, classification trees, and regression trees can be used with the data.

Table III ACPVRM Descriptive Statistics [13]

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
ACPRVF	64	2.30	64.30	13.4625	9.2661
ACPRVM	64	.90	99.90	10.2531	12.5751
Valid N (listwise)	64				

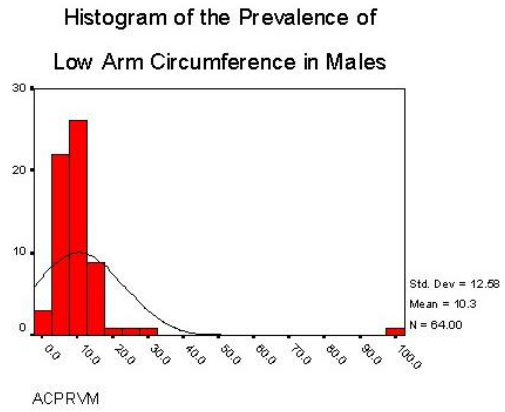

Fig. 1: ACPRVM With Dirty Data [13]

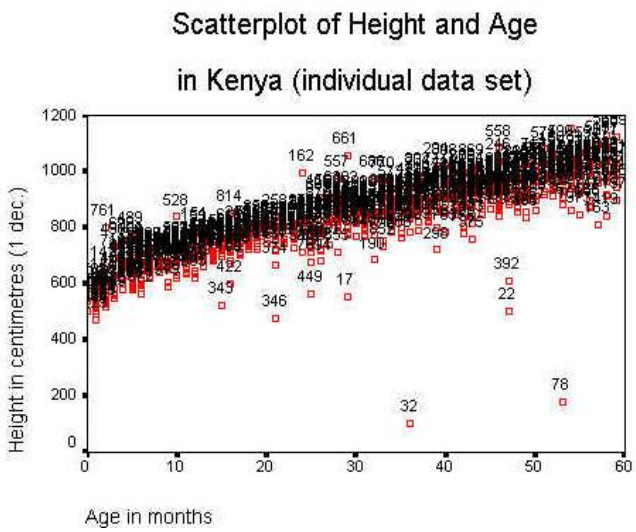

Fig. 2: Scatterplot With Outliers [13]

III. DATA MATCHING

In this section, we discuss the importance of data matching as an approach to data cleaning, and also discuss the process, and some results obtained by customizing the data matching process for use in a university environment.

A. Overview of Data Matching

With the rate at which man is generating data, either through his own actions (such as website visit data), or though actions of machines (such as aircraft instrument logs), the ability to analyse data in a timely manner can be of great benefit to businesses, governments, and even universities. Data matching seeks to identify and match records from separate databases. If records are not matched properly, even the most basic of statistical methods will give an inaccurate picture of the state of the environment the data was collected from. For example, for longitudinal census activities, census data collected periodically has to be matched, in order to keep track of how a population changes over time [5]. There are therefore several lists involved. After data matching, statistical analysis can be used to come up with valuable information such as frequencies, averages, and trends. Another example is the health sector [5]; one can get a very detailed and very valuable picture of an individual by integrating records from doctors, hospitals, health insurers, and pharmacies. This requires matching patient records from multiple databases. One can then study impact of treatments and diseases, by tracking the issues that crop up in a patient's life. In Australia, data matching was used to link records from ambulances, death registers, and hospital, and it was concluded that there is a need to install defibrillators in ambulances, which has saved many lives. Yet another example of application of data matching [5] is for national security, where undesirable elements have to be located by matching their records in financial transactions and online activities.

One more application area is in bibliographic databases [5]. In modern times, research is disseminated online via databases such as Springer Link, IEEE Xplore, Google Scholar, and the ACM Digital Library. Funding agencies and performance assessment units in universities also rely on such databases to evaluate research performance. Measures such as h-index calculate a numerical score which helps with these evaluations. Working with these databases can be problematic because sometimes two people with the same name and working in the same fields can exist in both publishing papers. Given two research publications published by an author with such a name, it is therefore difficult to count them as two authored by separate authors, or two authored by the same author. Table IV shows an example of a set of bibliographic records retrieved from an online database, for the same publication, but this is not immediately obvious.

Table IV Different Bibliographic Records for the Same Publication [5]

Kearns, M., Li, M., and Valiant, L: Learning boolean formulas, ACM 41 (1994a), 1298-1328
M. Kearns, M. Li and L. Valiant: Learning boolean formulas. Journal of the ACM, 41, 1994; 1298-1328.
M. Kearns, M. Li, and L. Valiant: Learning boolean formulae. Journal of the Association for Computing Machinery 41(6), 1995, pp. 1298–1328

B. Data Matching Process

The process involves pre-processing, indexing, record comparison, record pair classification, and evaluation of matching quality.

1) Pre-Processing: The first stage of data matching is pre-processing, which ensures the two lists are in the same format (same fields in both lists). This involves removing unwanted characters, such as periods and quotes; expanding abbreviations; segmenting data (by splitting addresses into plot and town, for example); verifying the correctness of attributes (by checking whether the region code exists, for example).

2) Indexing/Blocking: Data matching ordinarily requires comparing all items in a list A with all items in a list B. If there are M items in list A and N items in list B, data matching would need M*N comparisons. This is quite a large number if there are a million records in both lists, for example, making the process impractical. Indexing (also known as blocking) groups record pairs, such that comparison is only done between pairs in a group. This works because, for example, if we are matching house occupants in Botswana, we can use an attribute such as city, to ensure that we only need to compare house occupants in Gaborone with other house occupants in Gaborone, not with those in Mochudi. The attribute "city" is known as the blocking key. Careful selection of a blocking key could go a long way in reducing the time it takes to run the data matching process.

3) Record Comparison: Record pairs are evaluated for similarity. A similarity score could be real number between 0.0 and 1.0, for example. A higher score means higher similarity and vice versa. Attributes in the two lists, A and B, could be selected and used for this purpose, such that each record would be represented by a string. There are algorithms, such as those that fall in the category of template matching algorithms [14], [15], that can be used on strings, to score how many operations are required to convert "Goalathe" into "Gaolathe", for example. The more dissimilar the two strings, the more operations are needed

4) *Record Pair Classification:* Based on the scores for the record comparison operation, a record pair could be classified as either a match, non-match, or possible match. Various algorithms exist for coming up with models which carry out such a classification. This type of classification falls into the domain of machine learning; algorithms such as neural networks and decision trees can be used to implement this task. Figure 3 shows results for a classification activity.

Candidate pair	SimSum	Classification
(a1, b2)	3.71	Non-match
(a1, b3)	6.10	Match
(a1, b6)	3.39	Non-match
(a2, b2)	4.08	Potential match
(a2, b4)	2.56	Non-match
(a3, b1)	5.15	Potential match
(a3, b2)	2.91	Non-match
(a5, b4)	7.78	Match
(a5, b7)	3.07	Non-match
(a6, b5)	7.12	Match
(a7, b3)	2.98	Non-match
(a7, b6)	4.85	Potential match

Fig. 3: Three-class Classification Results [5]

5) Evaluation of Matching Quality: In order to evaluate a data matching process, metrics such as matching quality and matching completeness are used [5]. Matching quality refers to what proportion or reported matches are actual matches, while matching completeness refers to what proportion of actual matches are reported as matches. These are related to precision and recall metrics in standard statistical experiments. Complexity is also used: how many record pairs were generated by indexing compared to the number generated by a naïve approach which compares each record in the first list against all the records in the second list.

C. Data Matching Application

The authors applied data matching to the problem of processing ICT student marks in the University of Botswana (UB) [16]. ICT121 and ICT122 (Computing Skills Fundamentals I and II) are Computer Literacy courses which most first year students in UB enrol in. The final exam is multiple choice and is administered using special forms which an OMR scanner reads for automated marking. Students have to shade their details using an HB pencil. Mistakes are made when shading, so a student with student ID number "212312345" might shade it as "212312346". There is therefore a need to match records in the class registration list with those in the exam list to ensure the right student gets the right mark. The presence of other attributes makes this possible, namely Surname and Initials. So a pair of records, one from the registration list and one from the exam list, may still be flagged as a match, because the ID numbers are almost the same, while the Surname and Initials are identical, thus generating a template matching score of 0.99, when doing approximate string matching, while the record in question from the registration list has lower scores for all other comparisons.

The authors achieved very good results when evaluating the matching process on a database of 4289 student records. We introduced an additional pre-processing step to eliminate all perfect matches, before applying data matching. This led to reduction of 99.9% of naïve record pairs. The precision score was 1.0 (no misleading matches i.e. no false negatives). The recall score was also 1.0, so all matches were identified. Pairs quality was 1.0, meaning that all candidate record pairs generated correspond to true matches.

IV. MISSING DATA IMPUTATION

Missing data arise in almost all serious statistical analysis. Imputation is the process of replacing missing data with substituted values in order to increase accuracy. Some well-known methods of dealing with missing data include: list-wise and pairwise deletion, mean imputation, regression imputation, prediction methods, maximum likelihood, single and multiple imputation.

Many decision making processes use predictive models that take observed data as inputs. Such models break down when one or more inputs are missing [6]. Ignoring missing data can lead to unreliable results. Decision making tools such as those found in computational statistics (neural networks, support vector machines, etc.) cannot be used for decision making if data are not complete. Due to time constraints, undesirable approaches are often used, such as case deletion, impacting on quality of results. In many applications, there is a limited time between sensor readings, during which imputation has to be done, it has to be done quickly, therefore.

Insertion of the mean value for a variable is sometimes used. Sometimes zero is used. It is also possible to detect a pattern that governs missing data, it might be possible to identify cases and variables that affect the missing data, leading to a more reliable estimation method. Figure 4 shows different missing data patters: starting from the leftmost sub-figure, univariate and monotone (which could be caused by sensor failures) and arbitrary patterns (which is caused by data being recorded by different individuals, such as in a medical database).

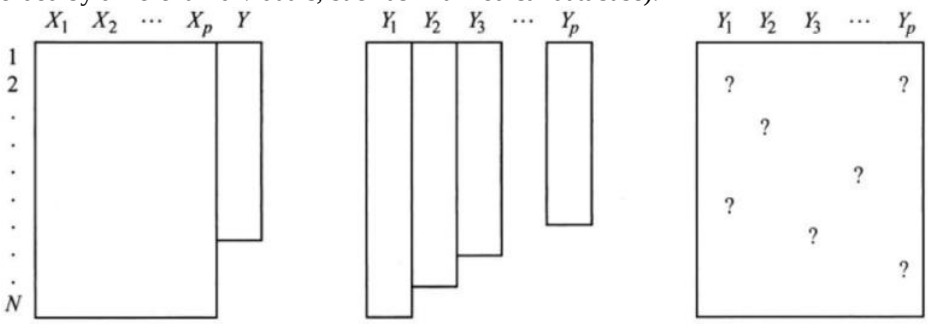

Fig. 4: Missing Data Patterns [6]

Since 2000, many new ways have been proposed for missing data imputation, including using neural networks, genetic algorithms, particle swarm optimization, pseudo-nearest-neighbour approach, decision trees. It has been found that when 2% of the features are missing, and the complete observation is deleted, up to 18% of the data might be lost. Figure 5 shows an example of an auto-associative neural network which maps input data values to identical values, in the process learning what valid data "looks like." [6]. Figure 6 shows how the auto-associative neural network (autoencoder) can be used with Genetic Algorithms. Records with good data and estimated data are fed into the auto-associative neural network. If the estimated data does not conform to the actual data distribution, the output will not be the same as the input, thereby allowing for an error to be calculated. The Genetic Algorithm, which is an optimization approach, will try to minimize this error by trying to come up with appropriate values for the missing data.

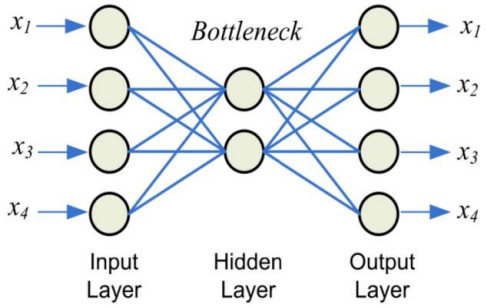

Fig. 5: Auto-Associative Neural Network [6]

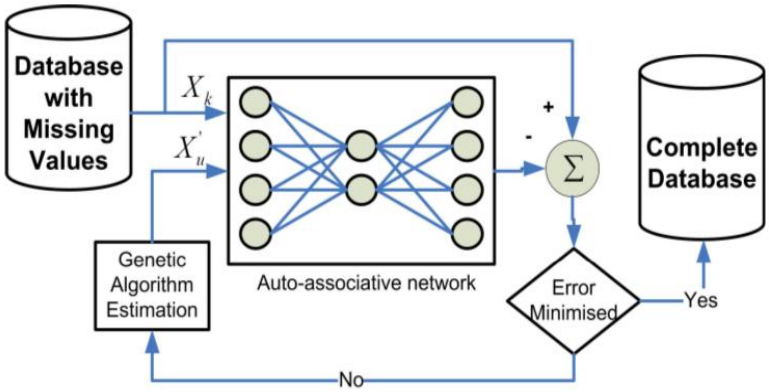

Fig. 6: Missing Data Imputation System Using an Auto-Associative Neural Network and Genetic Algorithms [6]

V. UNIFICATION OF DATA MATCHING AND MISSING DATA IMPUTATION

Data Matching can help with Missing Data Imputation when merging databases, reducing the number of records with missing data, enlarging the set of data that approaches such as an autoencoder and Genetic Algorithms can train from.

Missing Data Imputation can help with Data Matching by inferring values for data fields e.g. city based on postal code, or gender based on name, but a huge amount of data is required. We propose the Data Matching-Missing Data Imputation Data Cleaning Framework (DMMDIDCF). The flow chart for the framework is shown in Figure 7. This framework works in a scenario where two databases have to be integrated, where a database could be a list of records. The databases might need to be integrated as part of a data warehouse solution. DB1 and DB2 are subsets of the two databases, for which record pairs have already been identified, through a manual process, for example. However, they might still have some missing data. This is addressed by the missing data correction process. This might involve manually or semi-automatically replacing values for missing data. Rules are learned for how to handle missing data based on this process. For example, if missing data correction makes use of mean values for missing values belonging to a variable, this rule is learned and is used in the main Missing Data Imputation process, which is automated. After missing data correction, data matching models are developed, and these feed into the main data matching process, which will work on data for which users of the framework do not know correct matches. After data matching is executed, the data can now be used for statistical analysis, data mining, business intelligence, etc. If during this use anomalies are detected, there is feedback into the data matching training process, and the missing data correction process. Feedback could be identification that a value used in place of a missing value is incorrect, or an incorrect pairing of records was made during data matching. The feedback updates the data matching process and missing data correction process. DB3 and DB4 are subsets of the two main databases which are being integrated and therefore have to be cleaned. They make up the bulk of the records. They represent data which will not be processed manually or semi-automatically, but rather automatically.

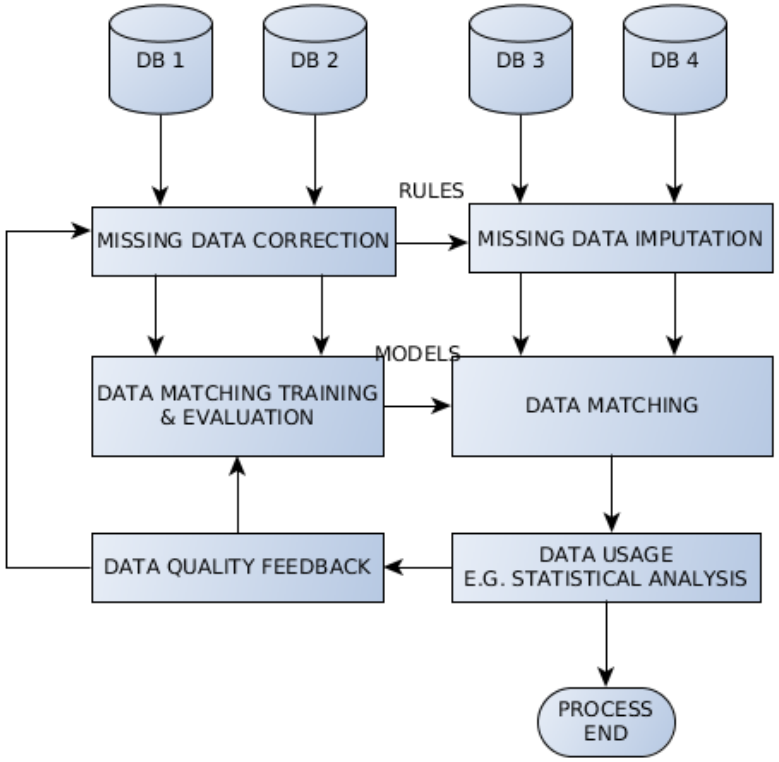

Fig. 7: Data Matching-Missing Data Imputation Data Cleaning Framework

To illustrate the usage of the framework, we use the example of integrating student databases in a university. A student's course record for course CSI133 (DB 2) could have fields for ID, Surname, Initials, Programme Code, Assignment 1, Assignment 2, Test 1, Test 2, CA, Exam Mark, Final Mark. During error correction, if some records are identified with a missing mark, such as Test 2, but the other marks are present, with knowledge of the way final marks are calculated, the Test 2 mark could be recreated. This formula will feed into the Missing Data Imputation module as a rule. DB 2 will be merged with DB 1. DB 1 could be coming from the registration list. The Data Matching Training and Evaluation module will try to anticipate errors in student ID in DB 2, ensure the record pairs from DB 1 and DB 2 are matched correctly. These models then feed into the Data Matching module. Once the Missing Data Correction and Data Matching Training and Evaluation rules are generated, then DB 3 and DB 4 can be processed, to deal with problems which must be solved automatically.

After data matching, statistical methods are used on the data, as part of the Data Usage module, to discover means of marks, frequencies to calculate pass rates, linear regression models to predict marks for students; various other statistical methods can be used as part of business intelligence or other applications. During the Data Usage phase, it might be detected that the highest final mark for a student is greater than 100, which might mean that Missing Data Imputation did not use an acceptable value for a missing mark. This will be used as feedback to Missing Data Correction, so that new, better, rules can be generated. During Data Usage, it might also be observed that a student the user is familiar with has an incorrect ID. This could indicate an incorrect match during Data Matching. This will then be used as

feedback to the Data Matching Training and Evaluation module, which will use this information to come up with a better classifier or scoring system for matches, for this type of data.

VI. CONCLUSION

In this paper, we describe data cleaning as a process used to determine dirty data and to correct detected errors and omissions in order to improve the quality of the data. We have shown that the data warehouse technology is a complex process involving integrating data from multiple sources that always results in creation of dirty data. We have demonstrated that dirty data contributes to inaccurate and unreliable results in decision making and once detected, it has to be corrected.

We discussed several existing data cleaning techniques and methods and focus on Data Matching and Missing Data Imputation. We design a framework that merges missing value imputation and data matching. This framework produces data which can then be used for various statistical analyses, both computational and conventional.

The next stage is to apply the framework to a data warehouse development process and measure the sensitivity of the selected statistical analysis methods (applications in Data Usage) to various data matching problems and missing data problems and demonstrate how this can improve data in a data warehouse for better decision making

REFERENCES

[1] W. L. Martinez and A. R. Martinez, *Computational Statistics Handbook with Matlab*, Boca Raton, USA: CRC, 2002.

[2] D. M. Diez, C. D. Barr, and M. Çetinkaya-Rundel, *OpenIntro Statistics*, 3rd ed., 2015. [Online] Available: http://www.openintro.org.

[3] R Core Team, *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing, 2015. [Online] Available: https://www.R-project.org.

[4] E. Rahm and H.H. Do, "Data Cleaning: Problems and Current Approaches," *IEEE Data Engineering Bulletin*, vol. 23, no. 4, pp. 3-13, 2000.

[5] P. Christen, Data Matching, Berlin, Germany: Springer-Verlag, 2012.

[6] F. V. Nelwamondo, "Computational Intelligence Techniques for Missing Data Imputation," PhD Thesis, University of the Witwatersrand, Johannesburg, South Africa, 2008.

[7] L. Baker, *Practical Data Cleaning*, Victoria: LeanPub, 2015.

[8] N. Choudhary, "A Study over Problems and Approaches of Data Cleansing/Cleaning," *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, vol. 4, no. 2, pp. 774-779, 2014.

[9] S. Rana, Er. G. P. Negi, and K. Kapoor, "A Study over Data Cleaning and Its Tools," *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, vol. 6, no. 3, pp. 553-558, 2016.

[10] S. Rana, Er. G. P. Negi, and K. Kapoor, "A Study over Importance of Data Cleansing in Data Warehouse," *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, vol. 6, no. 4, pp. 151-157, 2016.

[11] S. Chaudhuri and U. Dayal, "An Overview of Data Warehousing and OLAP Technology," *SIGMOD Rec.*, vol. 26 no. 1, pp. 65-74, 1997.

[12] Wikibooks (2015) Statistics. [Online]. Available: https://en.wikibooks.org/wiki/Statistics.

[13] Tulane University Medical School (n.d.) Practical Analysis of Nutritional Data. [Online]. Available: http://www.tulane.edu/~panda2/Analysis2/datclean/dataclean.htm.

[14] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys*, vol. 33 no. 1, pp. 31-88, 2011.

[15] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras, *Introduction to Pattern Recognition: A Matlab Approach*, Kidlington, Oxford, UK: Academic Press, 2010.

[16] G. Anderson, A.N. Masizana, and D. Mpoeleng, "An Exact and Inexact Data Matching Approach for Saving Time and Preventing Errors in Processing of Student Exam Results at the University of Botswana," *International Journal on Information Technology (IREIT)*, vol. 1, no. 3, pp. 179-185, 2013.

[17] E. de Jong and M. van der Loo, *An Introduction to Data Cleaning with R*, The Hague: Statistics Netherlands, 2013.

