Article

How to Address the Data Quality Issues in Regression Models: A Guided Process for Data Cleaning

David Camilo Corrales 1,2, *,†, Juan Carlos Corrales 3,† and Agapito Ledezma 2,†

1 Grupo de Ingeniería Telemática, Universidad del Cauca, Campus Tulcán, 190002 Popayán, Colombia
2 Departamento de Ciencias de la Computación e Ingeniería, Universidad Carlos III de Madrid, Avenida de la Universidad, 30, 28911 Leganés, Spain; ledezma@inf.uc3m.es
3 Grupo de Ingeniería Telemática, Universidad del Cauca, Campus Tulcán, 190002 Popayán, Colombia; jcorral@unicauca.edu.co
* Correspondence: dcorrales@unicauca.edu.co or davidcamilo.corrales@alumnos.uc3m.es; Tel.: +57-28209800 (ext. 2129)
† These authors contributed equally to this work.

Received: 9 March 2018; Accepted: 30 March 2018; Published: 6 April 2018

Abstract: Today, data availability has gone from scarce to superabundant. Technologies like IoT, trends in social media and the capabilities of smart-phones are producing and digitizing lots of data that was previously unavailable. This massive increase of data creates opportunities to gain new business models, but also demands new techniques and methods of data quality in knowledge discovery, especially when the data comes from different sources (e.g., sensors, social networks, cameras, etc.). The data quality process of the data set proposes conclusions about the information they contain. This is increasingly done with the aid of data cleaning approaches. Therefore, guaranteeing a high data quality is considered as the primary goal of the data scientist. In this paper, we propose a process for data cleaning in regression models (DC-RM). The proposed data cleaning process is evaluated through a real datasets coming from the UCI Repository of Machine Learning Databases. With the aim of assessing the data cleaning process, the dataset that is cleaned by DC-RM was used to train the same regression models proposed by the authors of UCI datasets. The results achieved by the trained models with the dataset produced by DC-RM are better than or equal to that presented by the datasets’ authors.

Keywords: data cleaning in regression models (DC-RM); data quality issue; data cleaning task; regression model

1. Introduction

The creation and consumption of data continue to grow by leaps and bounds. Due to advances in Information Technologies (IT), today the data explosion in the digital universe is a new trend [1,2]. The vast amount of data comes 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 to 1.7 Megabytes from 2020 [3].

Thus, knowledge discovery and data mining gain importance due the abundance of data [4]. A successful process of knowledge discovery is necessary to undertake data treatment. For example, a preliminary step in the knowledge discovery tasks is the data preprocessing, where the main goal is the cleaning of raw data [5]. From data mining, recognized methodologies have been proposed: Knowledge Discovery in Databases (KDD) [6], Cross Industry Standard Process for Data Mining (CRISP-DM) [7], Sample, Explore, Modify, Model and Assess (SEMMA) [8]. Each of these describes a phase for data preprocessing. However, these methodologies do not explain how to address in detail
the main issues in data cleaning, leaving out relevant analyses that may lead to problems related to poor data quality in data mining, machine learning, and data science projects [9].

To achieve a solution, the problems mentioned above, we propose a guided process for data cleaning in regression models (DC-RM). The procedure for building the process of data cleaning in regression models consist of identity, understand, organize and filter the data quality issues according to their meaning. After, for each data quality issues found in the datasets, a data cleaning task is suggested. Finally, we validate our approach applying the data cleaning process (DC-RM) to real datasets coming from UCI Repository of Machine Learning Databases [10], training the same algorithms used by the datasets authors with the dataset obtained by DC-RM, and comparing the precision achieved by these. The remainder of this paper is organized as follows: Section 2 discusses definitions of regression models, data quality and the related works. The process for data cleaning is explained in Section 3; Section 4 presents the results and Section 5 presents the conclusions and future works.

2. Materials and Methods

2.1. Background

This section exposes the concepts for building the process for data cleaning in regression models (DC-RM).

2.1.1. Regression Models

In knowledge discovery, regression models seek relations among a dependent variable (numeric) and a set of independent variables (numeric or nominal) through a learning process from a curated samples [11]. The main goal of the regression models is to obtain accurate predictions similarly to the values of the independent variable for new samples [12]. The Regression algorithms commonly used in the literature include multilayer perceptron [13], radial basis function network [14] and regression trees [15].

2.1.2. Data Quality

Data is affected by several processes, most of which affect its quality to a certain degree [16]. The authors in [17] define data quality as “the degree of fulfilment of all those requirements defined for data, which is needed for a specific purpose”. On the other hand, according to the authors of [18], the data errors may affect the predictive accuracy of linear regression models in two ways. First, the training data used to build the model may contain errors. Second, even if the training data are free of errors, once a linear regression model is used for forecasting a user may input test data containing errors to the model. This study demonstrates that the outputs of one linear regression model are sensitive to data errors. From the two assumptions above, this research is focused on the first one, taking GIGO: “Garbage In Garbage Out” as principle.

2.2. Related Works

Several researchers have built mechanisms to address data quality issues. Table 1 presents a summary of the related works and the data quality issues addressed.

In Table 1, we observe a large diversity of approaches for addressing data quality issues designed for relational data bases, data warehouses, health systems, and other domain applications (wind energy, seismic waves, electricity consumption); however, the related works are not focused on regression tasks of knowledge discovery. Although the following works [19,20] are oriented for big data pre-processing, they lack a user-oriented process to address orderly many data quality issues (e.g., missing values, outliers, duplicate instances, high dimensionality).
Table 1. Related works.

<table>
<thead>
<tr>
<th>Works</th>
<th>Domain</th>
<th>Data Quality Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>[16,21–26]</td>
<td>Databases</td>
<td>Integrity constraints, validation of overloaded table, duplicate records, inconsistencies, missing values, data freshness.</td>
</tr>
<tr>
<td>[9,27–30]</td>
<td>Conceptual</td>
<td>Incomplete data, out-of-date data, lack of meta-data, high time to understand data</td>
</tr>
<tr>
<td>[31–38]</td>
<td>Health systems</td>
<td>Data heterogeneity, illegible handwriting, missing values, duplication of patient records, inconsistency, unsuitable data format, timeliness, inaccuracy.</td>
</tr>
<tr>
<td>[39–41]</td>
<td>Wind energy, seismic waves, electricity consumption</td>
<td>Data heterogeneity, missing values, out-of-date data, noise, outliers, inconsistency</td>
</tr>
<tr>
<td>[19,20]</td>
<td>Big data</td>
<td>Missing values, timeliness, contextual adequacy, operational adequacy, temporal adequacy</td>
</tr>
</tbody>
</table>

3. Process for Data Cleaning in Regression Models

This section presents the process to address poor quality data in regression tasks. The methodology “Building a Conceptual Framework: Philosophy, Definitions, and Procedure” [42] was adapted to build the proposed process. This offers an organized procedure of theorization for building conceptual process. The advantages of using this methodology proposed by [42] are the flexibility for make modifications, and the easy understanding. The procedure for building the process of data cleaning in regression models consists of the following phases:

3.1. Mapping the Selected Data Sources

The first phase identifies the data quality issues to regression tasks. Data sources as research papers and methodologies were reviewed:

- From data mining and machine learning four relevant methodologies, we found: Knowledge Discovery in Databases (KDD) [6], Cross Industry Standard Process for Data Mining (CRISP-DM) [7], Sample, Explore, Modify, Model and Assess (SEMMA) [8] and The Data Science Process [43]. These methodologies mention data quality issues such as: missing values, outliers, duplicate instances and high dimensionality.

- In [44] we present a literature review for data quality issues in knowledge discovery tasks. We reviewed research papers from IEEE Xplore, Science Direct, Springer Link, and Google Scholar. Based on the literature analysis, it can be stated that three quality issues were found: missing values, outliers, and redundancy (refers to duplicate instances). Also, the noise was identified as a data quality issue (see Figure 1).

Data quality issues such as missing values, outliers and redundancy have received greater attention from research community (papers found: 39, 47 and 55 respectively). Meanwhile noise (17 papers) has been paid less attention because it is defined as general consequence of the data measurement errors.
Figure 1. Number of papers found for each data quality issue [44].

3.2. Understanding the Selected Data

The aim in this phase is understand the data quality issues from regression task. Next a description of each data quality issue is presented:

- **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 [45].
- **Outlier**: can be an observation univariate or multivariate. An observation is denominated an outlier when it deviates markedly from other observations, in other words, when the observation appears to be inconsistent respect to the remainder of observations [46–48].
- **High dimensionality**: is referred to when dataset contains a large number of features [49]. In this case, the regression model tends to overfit, decreasing its performance [50].
- **Redundancy**: represents duplicate instances in data sets which might detrimentally affect the performance of classifiers [51].
- **Noise**: defined by [52] as irrelevant or meaningless data. The data noisy reduce the predictive ability in a regression model [53].

3.3. Identifying and Filtering Components

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

- **Redundancy** were renamed as **Duplicate instances** to represent better the data quality issues in regression models.
- **Noise** is considered a general issue according it definition: “irrelevant or meaningless data”. Thus **Missing values**, **Outliers**, **High dimensionality** and **Duplicate instances** are considered as a kind of **Noise**.
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 (see Table 2).

Table 2. Data cleaning tasks in regression models.

<table>
<thead>
<tr>
<th>Noise Issue</th>
<th>Data Cleaning Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing values</td>
<td>Imputation</td>
</tr>
<tr>
<td>Outliers</td>
<td>Outlier detection</td>
</tr>
<tr>
<td>High dimensionality</td>
<td>Dimensionality reduction</td>
</tr>
<tr>
<td>Duplicate instances</td>
<td>Remove duplicate instances</td>
</tr>
</tbody>
</table>

- **Imputation**: replaces missing data with substituted values. In the literature we found four relevant approaches to imputing missing values:
  - *Deletion*: excludes instances if any value is missing [54].
  - *Hot deck*: missing items are replaced by using values from the same dataset [55].
  - *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) [56].
  - *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 [57].

- **Outlier detection**: identifies candidate outliers through approaches based on *Clustering* (e.g., DBSCAN: Density-based spatial clustering of applications with noise) or *Distance* (e.g., LOF: Local Outlier Factor) [58–60].

- **Dimensionality reduction**: reduces the number of attributes finding useful features to represent the dataset [61]. A subset of features is selected for the learning process of the regression model [49]. The best subset of relevant features is the one with least number of dimensions that most contribute to learning accuracy [62]. Dimensionality reduction can take on four approaches:
  - *Filter*: selects features based on discriminating criteria that are relatively independent of the regression (e.g., correlation coefficients) [62].
  - *Wrapper*: based on the performance of regression models (e.g., error measures) are maintained or discarded features in each iteration [63].
  - *Embedded*: the features are selected when the regression model is trained. The embedded methods try to reduce the computation time of the wrapper methods [64].
  - *Projection*: looks for a projection of the original space to space with orthogonal dimensions (e.g., principal component analysis) [65].

- **Remove duplicate instances**: identifies and removes duplicate instances [66].

Several data cleaning tasks were identified for regression models. The integration of the data cleaning tasks is depicted in Figure 2:
Below, we explain the step-step execution of data cleaning flow for the regression models.

1. **Verify if dataset contains missing values**: usually missing data are represented by special characters such as ?, *, blank spaces, specials words as NaN, null, etc. The first step is convert the missing values to format of the data cleaning algorithm.

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

3. **Apply outliers detection algorithm**: the outlier detection algorithm searches candidate outliers in the raw dataset or erroneous values generated by Imputation techniques.

4. **Apply algorithms to remove duplicate instances**: these algorithms search for duplicate instances in both the raw dataset or those generated by imputation algorithms.

5. **Apply algorithm for dimensionality reduction**: this kind of algorithms reduce the high dimensional in data sets by selecting a subset of most relevant features [67]. Different authors [68,69] assert that the feature selection methods have several advantages, such as: (i) improving the performance of the classifiers; (ii) better visualization and data understanding; and (iii) reducing time and computational cost

Once the phase of components integration is finished, we develop a software prototype of the guided process for data cleaning in regression models. Figure 3 presents the layer view of DC-RM. DC-RM is composed by four layers:

- **Graphical User Interface (GUI)** enables an user of DC-RM interact with the algorithms of data cleaning through graphical elements, such as text, windows, icons, buttons, text fields, etc. We developed two main forms in NetBeans IDE 8.2. 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 4.
The second form (it appears when the button “Start cleaning” of the first form is pressed) presents the algorithms for each data cleaning task and the DC-RM process. In Figure 5 is depicted the second form when the chi–squared algorithm is applied in the dimensionality reduction phase.

- Java code establish a connection with R through Rserver, subsequently it invokes the data cleaning algorithms of the R packages, and finally, it sends the results of data cleaning algorithms to Graphical User Interface.
- Rserve acts as a socket server (TCP/IP or local sockets) which responds to requests from Java code. It listens for any incoming connections and processes incoming requests [70]. In other words, Rserve allows to embed R code within Java code.
- R is a system for statistical computation and graphics. It provides a programming language as dialect of S which was designed in the 1980s and has been in widespread use in the statistical community since [71]. R methods are based on packages, they are collections of functions and data sets developed by the community. We used R version 3.4.2 with missForest package [72] for imputation task, Rlof [73] and fpc [74] packages for outliers detection task, and Fselector [75] package for dimensionality reduction tasks. In case of remove duplicate instances, we used R function duplicated().

Figure 3. Layer view of DC-RM.

Figure 4. Form of the statistical information of the dataset.
3.5. Validation

We evaluate the process for data cleaning in regression models (DC-RM) through real datasets coming from UCI Repository of Machine Learning Databases [10]. Section 4.2 explains in detail the validation of the proposed process for dataset: prediction of comments in Facebook post (cFp) [76].

4. Experimental Results

The DC-RM process was applied to real datasets of the UCI Repository of Machine Learning Databases [10]. Subsequently, the cleaned datasets by DC-RM are used to train the same algorithms proposed by authors of UCI datasets. Finally, we compare the Mean Absolute Errors (MAE) of the models trained with the datasets produced by the authors versus the models trained with the datasets processed by DC-RM.

The Mean Absolute Error (MAE) is defined by the next Equation:

$$\text{MAE} = \frac{\sum_{i=1}^{n} Y_i - \hat{Y}_i}{n}$$

where $Y_i$ is the actual measurement (comments in Facebook posts), $\hat{Y}_i$ is the predicted value and $n$ is the number of measurements.

We expose as case study the dataset for prediction of comments in Facebook posts (cFp) [76]. Thus, in Section 4.1 we presented the description of cFp dataset; in Section 4.2, we described the processing of the cFp dataset using DC-RM; Section 4.3 exposes the MAE achieved by the regression models (the trained by original cFp dataset versus the trained by DC-RM), finally Section 4.4 shows additional results of DC-RM with other datasets of UCI Repository [10].

4.1. Dataset Description

The dataset proposed in [76] is oriented towards the comments prediction in a Facebook post. The dataset is composed by a data test with 10,120 instances and five training sets as shown Table 3.

The dataset contains 53 attributes: 4 page features (page likes, page category, etc.), 30 essential features (comment count in last 24 and 48 h, etc.), 14 Weekday features (binary variables related with the date of Facebook post), and 5 other basic features (length of document, post share count, etc.).
Table 3. Instances of dataset for prediction of comments in Facebook posts.

<table>
<thead>
<tr>
<th>Data Training</th>
<th>Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variant 1</td>
<td>40.949</td>
</tr>
<tr>
<td>Variant 2</td>
<td>81.312</td>
</tr>
<tr>
<td>Variant 3</td>
<td>121.098</td>
</tr>
<tr>
<td>Variant 4</td>
<td>160.424</td>
</tr>
<tr>
<td>Variant 5</td>
<td>199.030</td>
</tr>
</tbody>
</table>

4.2. Evaluation Process

4.2.1. Imputation

After executing the first step in the execution flow, 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 [71]. As a result of this operation, the dataset presents missing values in three attributes. Therefore, we test two imputation approaches.

- Global imputation based on non-missing attributes: the main idea is fill the missing values by regression models. Missing attributes are treated as dependent variables, and a regression is performed to impute missing values [57]. The random forest algorithm [77] was used to fill the missing values. This method builds a model for each variable. Then it uses the model to predict missing values in the variable with help of observed values.

- Global imputation based on missing attribute: assigns the most frequent value of the attribute to a missing values. Commonly, a measure of central tendency is used for filling the holes [57]. In this case, the mean imputation was used [78].

Table 4 presents the Mean Absolute Error of the imputation methods.

Random Forest reaches low MAEs 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 4 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 6 presents the imputed (red line) and original values (black line) for attribute 6 (comments average in last 24 h of the data training—variant 1).

Table 4. Mean absolute error for imputation methods.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attribute Index</th>
<th>Random Forest</th>
<th>Mean Imputation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variant 1</td>
<td>6</td>
<td>0.011</td>
<td>97.749</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>0.001</td>
<td>1.752</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>0.017</td>
<td>0.237</td>
</tr>
<tr>
<td>Variant 2</td>
<td>15</td>
<td>0.009</td>
<td>287.652</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>1.214</td>
<td>18.624</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>5.86 × 10⁻⁴</td>
<td>35.803</td>
</tr>
<tr>
<td>Variant 3</td>
<td>22</td>
<td>0</td>
<td>26.665</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>0.004</td>
<td>0.203</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.003</td>
<td>6.546</td>
</tr>
<tr>
<td>Variant 4</td>
<td>13</td>
<td>3.6 × 10⁻⁵</td>
<td>126.782</td>
</tr>
<tr>
<td></td>
<td>49</td>
<td>0.010</td>
<td>0.233</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>2.47 × 10⁻¹³</td>
<td>4.896</td>
</tr>
<tr>
<td>Variant 5</td>
<td>12</td>
<td>3.09 × 10⁻¹³</td>
<td>7.172</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>0.135</td>
<td>54.445</td>
</tr>
<tr>
<td></td>
<td>52</td>
<td>0.006</td>
<td>0.223</td>
</tr>
</tbody>
</table>
In Figure 6, we 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 h of the data training—variant 2 is shown in Figure 7.

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

4.2.2. Outliers Detection

After obtaining the imputed values, according to the execution flow presented in Figure 2, 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) and clustering (Density-Based Spatial Clustering of Applications with Noise) approaches.

- Local Outlier Factor (LOF): is an algorithm for identifying distance-based local outliers [58]. With LOF, the local density of a certain point is compared with its neighbors. If the former is
significantly lower than the latter (with an LOF value greater than 1), the point can be in a sparser region than its neighbors, which suggests it be an outlier [79].

- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN):** searches clusters with regions of high and low density [80]. DBSCAN chooses an arbitrary unassigned object p from the dataset. If p is a core object, DBSCAN finds all the connected objects and these objects are assigned to a new cluster. If p is not a core object, then p is considered outlier object and DBSCAN moves onto the next unassigned object. Once every object is assigned, the algorithm stops [81].

Table 5 shows the candidate outliers detected by LOF and DBSCAN.

The clusters of outliers created by DBSCAN reach among 97 and 219 instances (Table 5); 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 5), 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 8 presents principal components PC1 and PC2 for data training—variant 5; 99.99% of the information contained in the data training are retained by the first two components. The outliers are labeled with “+” in red.

![Figure 8. Outliers for data training—variant 5.](image)
The candidate outliers detected by Local Outlier Factor (Table 5) were removed which can be erroneous observations generated in the imputation task.

### 4.2.3. Remove Duplicate Instances

We use the Standard Duplicate Elimination algorithm to detect duplicate instances [66]. 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 [82]. Table 6 shows the number of duplicate instances for each data training set (remove 312 duplicate instances).

### Table 6. Duplicate instances for each data training set.

<table>
<thead>
<tr>
<th>Data Training</th>
<th>Duplicate Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variant 1</td>
<td>8</td>
</tr>
<tr>
<td>Variant 2</td>
<td>21</td>
</tr>
<tr>
<td>Variant 3</td>
<td>59</td>
</tr>
<tr>
<td>Variant 4</td>
<td>88</td>
</tr>
<tr>
<td>Variant 5</td>
<td>136</td>
</tr>
</tbody>
</table>

### 4.2.4. Dimensionality Reduction

Considering that the datasets are large with 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 [83]. 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 [84]. Chi-squared and Information Gain were the methods used:

- **Chi-Squared:** is defined as sum of the squares of the differences of the independent and dependent variable divided by the dependent variable for every value [85]:

  \[
  X^2 = \sum_{i=1}^{n} \frac{(I_i - D_i)^2}{D_i}
  \]

  where \(I\) is the independent variable, \(D\) is the dependent variable and \(i\) is the ith value of the dataset.

- **Information Gain:** measures the expected reduction in entropy (uncertainty associated with a random feature) [86,87]. Given \(S_x\) the set of training examples, \(x_i\) the vector of ith variables in this set, \(|S_{x_i=v}|/|S_x|\) the fraction of examples of the ith variable having value \(v\) [88]:

  \[
  IG(S_x, x_i) = H(S_x) - \sum_{v=values(x_i)}\frac{|S_{x_i=v}|/|S_x|}{H(S_{x_i=v})}
  \]

  with entropy:
\[ H(S) = -p + (S)\log_2 p + (S) - p - (S)\log_2 p - (S) \]

\( p \pm (S) \) is the probability of a training example in the set \( S \) to be close to the value of the class.

Figure 9 shows 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).

4.3. Results

With the aim of assessing the data cleaning process, we use the dataset cleaned by DC-RM for training the same regression models proposed by the authors of cFp dataset [76]. Then, we compare the results of MAE obtained by the two approaches. The authors of [76] 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. Table 7 shows the mean absolute errors (MAE) of the models generated by dataset cleaned with DC-RM and the models proposed by the authors of cFp dataset [76]; the underlined values represent the best MAE overall achieved by the models using DC-RM and the authors proposal [76].

REP Tree was the model with the lowest MAE for DC-RM, as the authors proposed [76]. In contrast, the M5P tree of [76] (training with Variant 5) was the model with the highest MAE.
Overall, the training sets Variant 2, 3, 4, 5 cleaning by DC-RM achieve the best MAE in its models. In case of Variant 1, the authors’ proposal [76] reaches a best measure with a difference of MAE overall 0.92 respect to DC-RM.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Model</th>
<th>Var 1</th>
<th>Var 2</th>
<th>Var 3</th>
<th>Var 4</th>
<th>Var 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC-RM</td>
<td>MLP</td>
<td>34.55</td>
<td>31.31</td>
<td>35.19</td>
<td>38.59</td>
<td>55.17</td>
</tr>
<tr>
<td></td>
<td>RBF</td>
<td>31.09</td>
<td>31.85</td>
<td>30.12</td>
<td>29.81</td>
<td>29.69</td>
</tr>
<tr>
<td></td>
<td>REP</td>
<td>29.28</td>
<td>30.22</td>
<td>28.41</td>
<td>27.89</td>
<td>29.33</td>
</tr>
<tr>
<td></td>
<td>M5P</td>
<td>35.53</td>
<td>30.32</td>
<td>32.68</td>
<td>50.77</td>
<td>32.59</td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td>32.61</td>
<td>30.92</td>
<td>31.60</td>
<td>36.76</td>
<td>34.19</td>
</tr>
</tbody>
</table>

Table 7. Mean absolute error obtained by: DC-RM and [76].

### 4.4. Comparative Study

In order to demonstrate the performance of guided process for data cleaning in regression models, DC-RM also was validated with datasets coming from UCI Repository of Machine Learning Databases [10]. The models used were Support Vector Regression (SVR), Linear Regression (LR), Random Forest (RF), M5P Decision Tree, and Multi Layer Perceptron (MLP). Table 8 shows the MAE of models produced by the authors of UCI datasets compared with the same models trained with the datasets processed by DC-RM. The values underlined in Table 8 correspond to the MAE’s lowest.

Once the UCI datasets are cleaned by DC-RM, 81.81% of the models reach a lower Mean Absolute Error than models proposed by the dataset’s authors. For the remaining 8.19% of the models, the authors proposal of the datasets: “Turbine decay” and “Energy uses of appliances” achieve lowest MAE. However, the MAE’s difference of the models generated by the authors with respect to models built with datasets processed by DC-RM is slight. In the case of “Turbine decay” dataset, the MAE’s difference of SVR models is 0.002 and 0.06 for the “Energy uses of appliances” dataset, using the RF models.

Compared with effort in data preparation and previous domain knowledge by dataset authors, DC-RM offers a general data cleaning solution for any application domain. DC-RM reaches or overcomes the results proposed by the dataset’s authors.

Table 8. Mean absolute errors of the models processed by DC-RM and datasets authors of UCI repository.
Table 8. Cont.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Ref.</th>
<th>Approach</th>
<th>Model</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Posts in Facebook pages</td>
<td>[92]</td>
<td>DC-RM</td>
<td>SVR</td>
<td>25.26</td>
</tr>
<tr>
<td>Authors</td>
<td></td>
<td>Authors</td>
<td>SVR</td>
<td>26.9</td>
</tr>
<tr>
<td>Feedback Blogs Prediction</td>
<td>[93]</td>
<td>DC-RM</td>
<td>M5P</td>
<td>5.70</td>
</tr>
<tr>
<td>Authors</td>
<td></td>
<td>Authors</td>
<td>M5P</td>
<td>6.06</td>
</tr>
<tr>
<td>I-Room temperature</td>
<td>[94]</td>
<td>DC-RM</td>
<td>MLP</td>
<td>0.47</td>
</tr>
<tr>
<td>Authors</td>
<td></td>
<td>Authors</td>
<td>MLP</td>
<td>1.13</td>
</tr>
<tr>
<td>II-Room temperature</td>
<td>[94]</td>
<td>DC-RM</td>
<td>MLP</td>
<td>0.34</td>
</tr>
<tr>
<td>Authors</td>
<td></td>
<td>Authors</td>
<td>MLP</td>
<td>0.88</td>
</tr>
<tr>
<td>I-Dinning room temperature</td>
<td>[94]</td>
<td>DC-RM</td>
<td>MLP</td>
<td>0.43</td>
</tr>
<tr>
<td>Authors</td>
<td></td>
<td>Authors</td>
<td>MLP</td>
<td>0.89</td>
</tr>
<tr>
<td>II-Dinning room temperature</td>
<td>[94]</td>
<td>DC-RM</td>
<td>MLP</td>
<td>0.32</td>
</tr>
<tr>
<td>Authors</td>
<td></td>
<td>Authors</td>
<td>MLP</td>
<td>0.78</td>
</tr>
</tbody>
</table>

5. Conclusions and Future Works

In this work, a process to address the data quality issues in regression models is proposed. From DC-RM, we conclude:

- DC-RM provides support to methodologies from data mining and machine learning. For instance, in Knowledge Discovery in Databases, DC-RM can support the Preprocessing and Data Cleaning, Data Reduction, and Projecton phases. In Cross Industry Standard Process for Data Mining, DC-RM gives support to Verify Data Quality and Clean Data steps. (especially: Sample, Explore, Modify, Model and Assess in Modify phase); and, in Data Science Process into the Clean Data phase. Finally, we would like to emphasize that none of the methodologies discussed above explain in detail how to address the data quality issues in regression models.

- The DC-RM approach reduces the time and effort invested by the user in pre-processing phase, since it detects the data quality issues and advises about the suitable approach and the execution order for data cleaning tasks.

- Once DC-RM has been used in the UCI datasets, the models reach or overcome the results compared by the models built for the dataset's authors [76,89–94]. DC-RM offers a general data cleaning solution for any application domain.

- The dimensionality reduction is an important task that must be applied in large feature space. Considering the high data dimensionality of the dataset proposed by [76], the filter methods were used in DC-RM (due their fast and low computational cost). However, several authors declare that other methods with high computational cost such as wrapper and embedded methods can obtain better results [49,64,95–97].

As future works, we propose:

- Building other processes for data cleaning in knowledge discovery tasks as classification and clustering.

- Including ontologies of specific domains to support some data quality issues; e.g., selection of relevant attributes based on expert knowledge. In the cancer domain, the ontology developed by [98] can be used for selecting the relevant attributes and avoid the use of algorithms with high computational complexity in dimensionality reduction tasks.

- Creating a case based reasoning (CBR) system for supporting the data cleaning process. The CBR will automatically recommend the suitable data cleaning algorithm (e.g., in outliers detection, the CBR suggests the local outlier factor algorithm to the user).
Acknowledgments: The authors are grateful to the research groups: Control Learning Systems Optimization Group (CAOS) of the Carlos III University of Madrid and Telematics Engineering Group (GIT) of the University of Cauca for the technical support. In addition, the authors are grateful to COLCIENCIAS for PhD scholarship granted to MsC. David Camilo Corrales. This work has been also supported by the Spanish Ministry of Economy, Industry and Competitiveness (Projects TRA2015-63708-R and TRA2016-78886-C3-1-R).

Author Contributions: This paper is the result of PhD thesis of David Camilo Corrales with the support of his supervisors Juan Carlos Corrales and Agapito Ledezma.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

DC-RM Data Cleaning in Regression Models

References


© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).