On Studying the Effect of Data Quality on Classification Performance

Roxane Jouseau, Sébastien Salva, and Chafik Samir

UCA - LIMOS, France

Abstract. During the last decade, data have played a key role for learning and decision making models. Unfortunately, the quality of data has been ignored or partially investigated as a pre-processing step. Motivated by applications in various fields, we propose to study data quality and its impact on the performance of several learning models. In this work, we first introduce a list of elementary repairing tasks ranging from easy to complex with an increasing level. Then, we form categories from the state-of-the-art cleaning and repairing methods. We also investigate if it is always efficient to repair data. By including standard classifications models and public dataset, our work enables their use in different contexts and can be extended to other machine learning applications.

Keywords: Data quality · Data engineering · Data cleaning · Data repairing · Classification · Machine learning.

1 Introduction

The field of data cleaning and data repairing is very active. The literature is rich in methods and tools to tackle data cleaning tasks, e.g: HoloClean [14],CleanML [10], ZeroER [16], BoostClean [9]. Confronted with this plethora of approaches, data scientists have to choose which repairing method to use, and this choice is complex. Indeed, they have different execution times and effectiveness. Moreover, choosing a method solely according to those criteria overshadows another question: what does using a method requires? Some approaches require complex metadata that should be accounted for since the production of this metadata takes part in the repairing process, if it was not previously available. Data scientists are not always aware of the difficulties of using data repairing tools or if they perform better enough to justify the extra work that is necessary to produce the required metadata. The amount of errors in the data also has an impact, as repairing a dataset with only a few erroneous entries would be approached differently than repairing a severely degraded dataset. In some cases, the data is too deteriorated to be exploited, and we only discover this information after spending time and effort repairing it.

All these factors lead us to this question: Is it always better to repair data? This paper aims to partially answer this question, by focusing on the case of unstructured and structured numeric datasets in the context of classification tasks. We tackle the question through five criteria:

- C1: the perceived difficulty of using a method according to experts. This is a piece of crucial information when choosing a repairing method as it can lead to spending a long time working on metadata.
- C2: the impact of the degradation of data, C2 investigates how classification models perform with untreated errors in comparison to repaired data. This comparison will be helpful in identifying how repairing methods improve the accuracy and f1 score of classification tasks. Moreover, it studies how different levels of degradation affect the results of classification
- C3: the effectiveness of the repairing tool. C3 aim to observe the repairing effectiveness of repairing tools. Moreover, it compares the difference in effectiveness between repairing methods that are simple to use and complex ones.
- C4: the impact of the type of error present in data. This allows us to identify differences in accuracy and f1 scores of classification methods with different types of error in training data. This information is valuable for the decision-making process of repairing the data.
- C5: the impact of the classification model used. C5 aims to verify whether the accuracy and f1 score of classification models are affected similarly by data errors.

Other work related to this problematic are: CleanML: A study for Evaluating the Impact of Data Cleaning on ML Classification Tasks [10] which investigates the impact of data cleaning on classification tasks, Data Cleaning: Overview and Emerging Challenges [6] which proposes a taxonomy of the data cleaning literature, and Detecting Data Errors: Where are we and what needs to be done? [1] which investigates whether data cleaning tools are robust enough to capture most real world errors.

We restrict our work to 5 types of errors we found in the literature: missing values ([10,14]), exact and partial duplicates ([10,16,8]), domain value violations ([9]) and outliers ([10,8]). To study these criteria we create deteriorated datasets by injecting data errors uniformly in clean datasets. Artificially injecting errors allow us to control the quantity of errors in data, their type, and to have access to a reference version to compare results.

In this study, we selected repairing methods based on the type of data they target, the error types to repair, and the availability of the metadata needed to use them on the datasets we chose. For missing values, the choice of the algorithm is very dependent on the repairing approach we want to take as the detection is trivial. A simple approach would be to impute the values with statistical data such as the median (R_med) or average value (R_mean) of the attribute or by taking into account the correlation between the dataset attributes (R_correl) [10]. For the repairing of duplicated data, we consider exact or partial copies. Datasets with keys or not, if datasets have primary keys, using a key-collision (R_key) method is a simple solution [10]. Otherwise data need to be compared for equality. For partial duplicates, it's more complex as we first need to define a measure of when two records are considered partial duplicates before trying to find them in the data. The definition of a threshold for partial duplicates is case-specific, but tools like ZeroER (R_ZER) [16], which we use in this study, can detect them. For outliers, we considered 4 methods: the standard deviation (R_std), the interquartile range (R_quart), the interquantile range (R_quant) and the data linter (R_linter) [10,8]. For domain value violations, a combination of regular expressions and types checks is usually used to detect this type of error (R_check) [9].

Since this study focuses on classification tasks, we have selected several classification models through scikit-learn [12]: Logistic regression (Cl_LR), K-Nearest Neighbors (Cl_KNN), Decision tree (Cl_tree), Random forest (Cl_rdForest), Ada boost (Cl_AdaB), Naive Bayes (Cl_NB), XGboost (Cl_XGB), Support vector classification (Cl_SVC), Gaussian process (Cl_GP), Multi-layer perceptron (Cl_MLP), Stochastic gradient descent (Cl_SGD), and Gradient boosting (Cl_GB). We chose these models in order to cover multiple approaches to classification tasks.

To answer the question "Is it always better to repair data?", we first present the context and scope of our work. Then, we study C1 and propose a method to evaluate how difficult is a repairing method to use. Secondly, the criteria C2 to C5 are studied through an experiment that allows the observation of the impact of data cleaning and repairing on classification tasks and analyze its results. These criteria allowed us to identify two categories of error type: low impact and high impact on accuracy and f1 score. Moreover we also observed that repairing methods perform similarly both at very low and very high levels of errors (10% and 80%) but differ in between. This experiment was conducted on datasets adapted to classification tasks that have very low percentages of errors. They also cover very different domains and have different dimensions and sizes. Last but not least, they are free to use, and are also used in many papers and hence increase the reproductibility of our results.

The paper is organized as follow: Section 2 investigates C1 and proposes an evaluation of the difficulty of using a repairing method. Section 3 presents the experiment we designed with its empirical setup and the analysis of the results corresponding to C2, C3, C4, and C5. Finally, we discuss the possible threats to validity and our results in Section 4 and conclude in Section 5.

2 C1: The perceived difficulty of using a method according to experts

Repairing methods are traditionally evaluated and compared using effectiveness (their ability to produce the desired outcome), accuracy (how close they are to the ideal outcome), and performance (how fast they return results). However, this gives an incomplete comparison. For instance, some tools require more complex metadata inputs than others ([7,14] etc). The time and efforts put into creating this metadata are often complicated to quantify and therefore disregarded. That is one of the reasons why data scientists rarely use them in the industry.

To account for this, we propose an evaluation process, which breaks down the repairing methods discussed in the introduction into steps and sub-steps, until we obtain elementary tasks (i.e. small actions difficult to split) describing

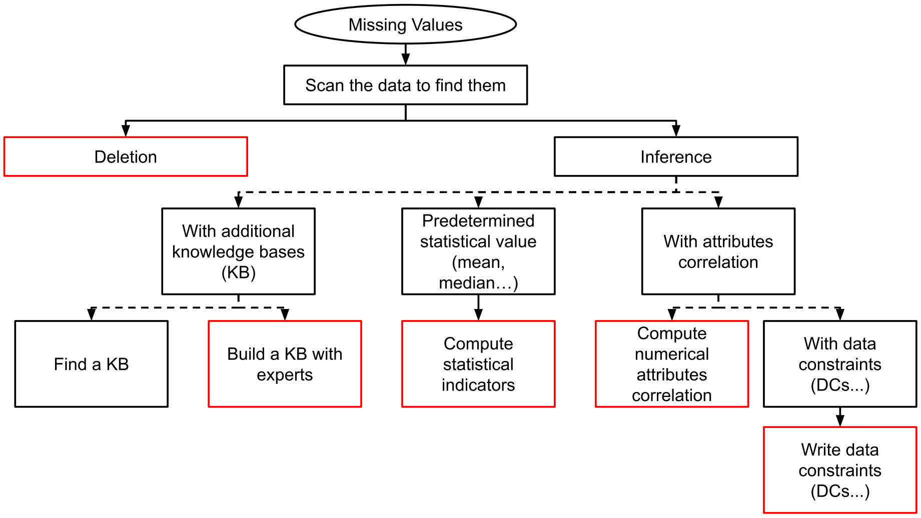
Missing Values
Scan the data to find them
Deletion
Inference
With additional knowledge bases (KB)
Predetermined statistical value (mean, median…)
With attributes correlation
Find a KB
Build a KB with experts
Compute statistical indicators
Compute numerical attributes correlation
With data constraints (DCs...)
Write data constraints (DCs...)

Fig. 1: Elementary tree decomposition for missing values

the actions executed to apply these methods, including creating the metadata needed. Given an error type e, and a repairing method R_{ei}, we build a tree expressing the steps of R_{ei}. For any other repairing method R_{ej} we complete the same tree with new sub-steps and branches when required. The final nodes of the tree are elementary tasks. For example, Figure 1, illustrates the tree achieved for the error type missing values with regards to the repairing methods [7,14,10,5]. These elementary tasks (nodes in red) are then evaluated by experts in terms of difficulty to complete them independently of the complete approaches. These evaluations allow us to finally compute a difficulty score for every repairing method as a summation of its elementary task evaluations.

Table 1: Difficulty ratings of the elementary tasks of data repairing.

Elementary tasks	Estimated difficulty (Easy, Medium, Medium +, Hard)
Compute statistical indicators	1.89
Delete data	1
Mining regexp	2
Mining data constraints	2.38
Write data conversion scripts	1.63
Compute attributes correlation	1.75
Write data harmonisation scripts	2.44
Define similarity metrics	2.88
Data scientist check the data (for miscoding)	2.13
Write data format rules (regexp...)	2.44
Write data constraints (DCs...)	2.71
Build a knowledge base with experts	3.33
Set a threshold (for partial duplicates detection)	2.67
Write a probabilistic model	3
Define a metric (for outliers detection)	3.56

To quantify the difficulty of each elementary task, we asked a panel of 8 industry data scientists to rank them on a four values scale: easy, medium, medium+, and hard. We chose a scale with four values to avoid having answers in the middle. We compute the difficulty score of an elementary task dt_t as a weighted arithmetic mean over the difficulty rankings given by the data scientists panel Table 1. $dt_t = \sum_{v=1}^{n} \frac{\delta_t(v)}{n}$, with $1 \leq dt_t \leq 4$, and with n the number of data scientists, and $\delta_{task}(v) = 1$, if the data scientist v ranked the elementary task t easy, 2 if medium, 3 if medium+ and 4 if hard.

We then compute the difficulty score of a repairing method dm_k using the difficulty scores of the elementary tasks included in the method. $dm_k = \frac{\sum_{i \in M} dt_i}{\sum_{j \in T} dt_j}$, with $0 \leq dm_k \leq 1$, and with T the set of all elementary tasks, and M the set of all elementary tasks used by the repairing method k. Table 2 illustrates the scores obtained for the repairing methods introduced in the introduction.

Table 2: Difficulty scores of repairing methods.

Repairing method	Elementary tasks included	Total estimated difficulty score
R_med, R_mean	Compute statistical indicators	0.053
R_correl	Compute statistical indicators, compute attributes correlation	0.102
R_key	delete data	0.028
R_ZER	Set a threshold for partial duplicates detection	0.075
R_std, R_quart, R_quant	compute statistical indicators, define a metric for outlier detection, delete data	0.18
R_linter	compute statistical indicators, delete data	0.081
R_check	write data conversion scripts	0.046

Table 1 shows that there is clear disparities in the perceived difficulties of using repairing methods. For instance R_correl is more complex than R_mean or R_med according to experts. We are now able to use these evaluations to compare two repairing methods before using them. This is especially useful in cases where we expect the two repairing methods to have similar effectiveness. We develop more on this subject in the following section.

3 Study of C2 to C5

Below, we study the criteria C2 to C5 presented in the introduction through an experiment on deteriorating and repairing numerical data. Due to space limitation we will only present a brief summary of our conclusions for C4 and C5. A technical report with additional figures is available [15].

3.1 Empirical setup

The datasets used are adapted to classification tasks and have very low percentages of errors. They are all numerical with different properties to cover a large panel of applications. Some are structured, some are not structured, some are from *real-world* data, whereas others are curated. They also come in various sizes. The datasets included are: Mnist, Fashion-Mnist, Olivetti, Iris, Adult, Breast cancer, and Wine [12,17,3]. We decided to limit the global computing time to under a week for each dataset. For this reason, we do not use the complete datasets for Fashion-mnist and Adult, but reduced versions of them (700 entries for Fashion-mnist and 2000 for Adult).

In this experiment, we start by splitting datasets into two: training and test. The training is subject to these modifications: We first inject the training dataset with one type of error e at a percentage p varying from 0 to 95% with increments of 5%. We apply each repairing method R_{ei} to different copies of the deteriorated dataset to obtain repaired datasets. We then use these repaired datasets to train several classification models Cl_X. Finally, we compute the accuracies and f1-scores by means of the testing sets. We executed the complete process 30 times to reduce the bias for each percentage p. We summarize all the steps of the experiment in Figure 2.

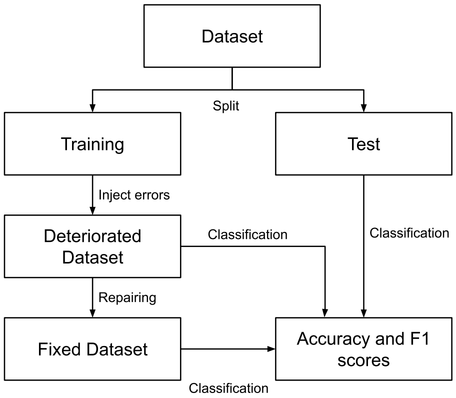

Fig. 2: Structure of the experiment

We use the errors, repairing methods and classification models presented in the introduction.

Our experiment aims to study C2 to C5: impact of the degradation of the data, effectiveness of the repairing tool, impact of the type of error, and impact of the classification model.

The following sections only show the experimental results related to the accuracy for space reasons but the experimental results for the f1 scores follow those of the accuracies.

3.2 C2: Impact of the degradation of the data on repairing effectiveness

In order to observe the impact of the degradation of data on the effectiveness of the repairing methods for each type of error, we randomly injected increasing percentages of errors in the datasets from 0 to 95% over the total amount of data. We trained the classification models on these data before and after repairing. In Figure 3, we respectively depict the mean accuracies of the classification models before and after repair, on all the datasets by error type as a function of the percentage of errors injected in the data. From Figure 3 (left), we identify two

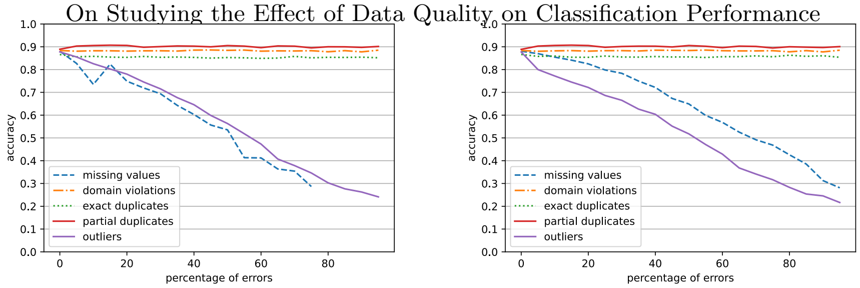

Fig. 3: accuracy score before (left) and after (right) repairing

distinct categories of error types: 1. the data degradation level has little to no impact on the accuracy, and 2. the data degradation level seems to have a big impact. The error types: domain value violations, exact duplicates, and partial duplicates belong to the first category, while missing values and outliers belong to the second category. Therefore, the impact of repairing is more interesting to observe for the second category. We also note a slight improvement of the accuracy for partial duplicates. This improvement actually is only observed for two datasets: wine and breast cancer [12]. It is caused by the fact that in this scenario creating partial duplicates act as enriching the data. For the other datasets, data is denser this is why adding partial duplicates has no effect on them. Domain value violations belonging to the first category can be explained by the fact that we need to repair domain value violations since we can't use an attribute with mixed data types for learning. From Figure 3 (right) we observe the same categories as we did before repairing the data. However the repairing of outliers seems to be less effective than the repairing of missing values. Since outliers are defined relatively to the rest of the data the repairing methods R_std, R_quart, R_quant and R_linter were defined statistically dependant to the data. They therefore become very quickly undetectable since having a lot of outliers populate the extremes of data values and they stop being distant from the data. The notion of a large amount of outliers is still relatively small for example, if 10% of data values are outliers, this is a significant number and cannot be considered distant from the data anymore.

We can clearly see that the impact of the degradation on data is very different depending on which category of error we observe. For the first category, the degradation of the data have very little impact on accuracy and so does repairing errors. However in the second category the degradation of the data has a strong impact on the accuracy. For example, outliers very quickly become too many to be able to detect them with simple statistical indicators. Repairing the missing values works better for higher percentages of degradation but we can see that we fall under a mean accuracy of 0.8 after 30% of missing values present in the data.

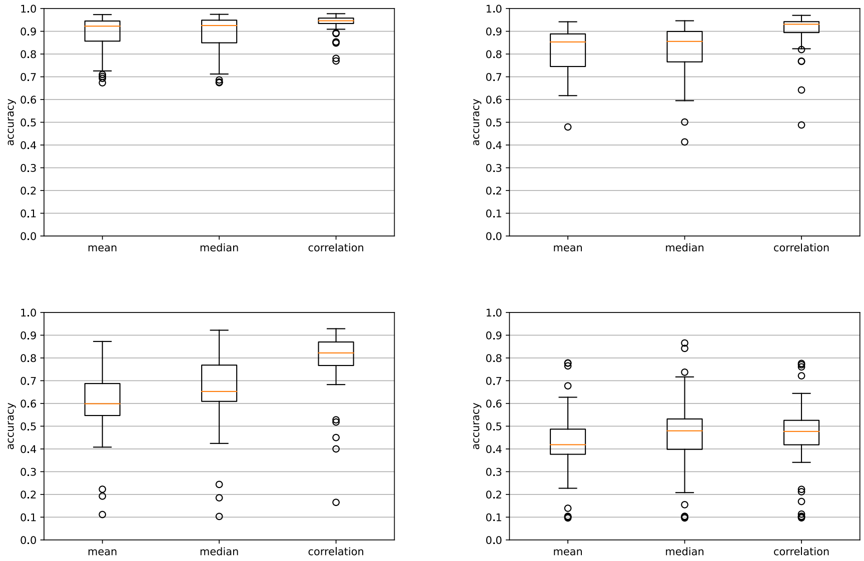

Fig. 4: effectiveness of some repairing tools for 10% (top left), 25% (top right), 50% (bottom left), and 75% (bottom right) of missing values.

3.3 C3: effectiveness of the repairing tools

To study the effectiveness of the repairing tools considered in the paper, we randomly injected errors at increasing percentages from 0 to 95% in our training sets. We then repaired them with different repairing methods, trained the classification models on those datasets and computed their respective accuracies and f1-scores. We only show the repairing of missing values and outliers in Figures 4 and 5 as their degradation has the most impact on accuracy as we saw in Section 3.2. Figure 4 shows the accuracies for 10%, 25%, 50%, and 75% of missing values after repairing with the methods R_med, R_mean, and R_correl. Figure 5 depicts the accuracies for 10%, 25%, 40%, and 50% of outliers after repairing with the methods R_std, R_quart, R_quant, and R_linter.

For missing values, with Figure 4 we observe that the repairing method using attributes' correlation is more efficient than imputation with the mean or median before 75% of missing values but performs similarly afterward. Repairing with R_mean or R_med seems to give very similar results before 50% of missing values but starting from 50% R_med gives slightly better results.

The choice of the repairing method, in this case, cannot be based solely on effectiveness as the methods perform more and more similarly as the percentage of missing values increases. Assessing an approximate percentage of errors in the dataset first and choosing a repairing tool afterward seems to be a good strategy in terms of efficiency and effectiveness (use of the simplest tool for the best outcome).

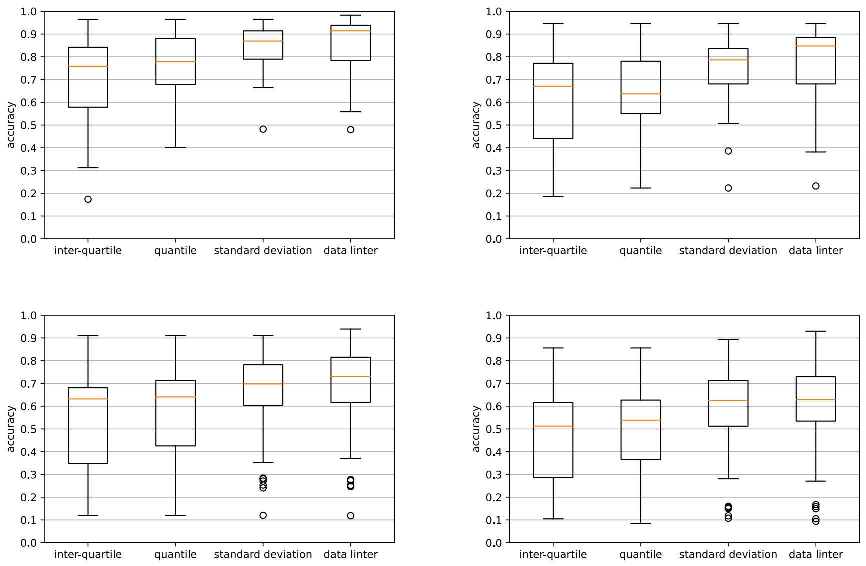

Fig. 5: effectiveness of some repairing tools for 10% (top left), 25% (top right), 40% (bottom left), and 50% (bottom right) of outliers.

For outliers, (Figure 5) the effectiveness of the different repairing tools seems to be less distinct than for missing values. However, the inter-quartiles and quantiles approaches seem to perform similarly, which is not surprising as they have similar concepts. Around 50% of degraded cells and up, none of the repairing methods we studied seem to perform well in terms of accuracy. This makes sense as outliers are, by definition, entries distant from the rest of the data. If there is a significant amount of them, they are not statistically distant anymore and thus undetectable with statistical indicators such as the standard deviation, quantiles, or quartiles. With high levels of outliers, repairing tools become less and less effective and their interquartile ranges seems to grow.

Overall, our experiment shows that the repairing methods does have an impact on the accuracy, some methods do perform better than other on average, but at high levels of degradation the effectiveness of the different repairing methods seems to be leveled out. For very low percentages of degradation of the data (10% and under) most methods perform well. But at higher levels of degradation (75% and up for missing values and 50% and up for outliers) most methods perform equivalently poorly. So depending on the accuracy we aim to reach, a relatively simple method could possibly give the desired results.

3.4 C4 and C5: impact of the type of error and impact of the classification model

In this section, we present a brief overview of our conclusions for C4 and C5, we do not detail these criteria for the sake of space.

The study of C4 shows that the type of error has a strong impact on the accuracies and f1 scores. We identified again the two categories of errors highlighted in Section 3.2. C4 also shows that outliers are the most complicated type of error to repair as they quickly become statistically significant and are thus harder to detect. Moreover partial duplicates should not always be repaired as we risk deleting non-duplicated data, they can even be considered as data enrichment in some cases.

The study of C5 shows that even for low percentages, the impacts of outliers on the classification models' are not equivalent for all models both before and after repairing. This difference grows as the percentage of outliers increases. But only up to a certain point as around 80% of degradation and up the models all tend to offer similar accuracies. We can therefor deduce that the choice of the classification model has an impact even after repairing, especially in the presence of outliers and in the presence of a high proportion of missing values.

4 Discussion

4.1 Is it always better to repair data?

Our work studied the impact of repairing methods on classification tasks to answer this question. We investigated it through five criteria. From them, we observed that it is interesting to repair data when the errors detected in the data are missing values and outliers (C2 and C4). Besides, C3 showed that for 10% of deterioration or less of the data, simple repairing methods (evaluated with C1) gave similar results to the one that can be obtained with more complex repairing methods. Hence, repairing data in this case tends to not bring any difficulty and is strongly recommended. C5 showed that it is interesting to repair outliers when these classification models Cl_AdaB, Cl_NB, Cl_SGD, Cl_XGB, and Cl_GP are used. On the contrary, data should not be repaired when partial duplicates are similar to data enrichment and hence artificially improve accuracies and f1 scores (C2 and C4). C2 and C3 also showed that at high degradation levels (starting from 75% for missing values and 50% for outliers), the repairing methods did not allow to improve accuracies and f1 scores. Hence using them at these degradation levels seems to be pointless. For the other situations, the decision of repairing data depends on several factors and has to be evaluated by data scientists. These factors correspond to the data scientist skills, the time they have to perform this task, if a complex repairing method using metadata is required, the percentage of errors present in the data, and the classification model used. We believe that our criteria may help them in the decision of repairing data.

4.2 Threats to validity

In this section, we address the internal and external threats to the validity of our work. We identified four internal threats: 1. The number of experts who answered the survey to rank the elementary tasks (Section 2), 2. The implementation of the classification models, 3. The hyper-parametrization of the classification models, 4. The parameters we chose for the repairing tools. To limit the impact of these internal threats the criterion C1 was studied by 8 data scientists working in the industry and experts in data repairing and machine learning. We implemented the classification models using scikit-learn [12], a widely used library. We hyper-parameterized the classification models on the original datasets using a grid search in order to avoid poor performance that can be caused by bad parametrization. Finally, for the repairing tools we chose widely used parameters such as the standard deviation, minimum and maximum or quantiles.

We also identified four external threats: 1. The choice of datasets, 2. The choice of the classification models, 3. The evaluation of the difficulty to do elementary tasks, and 4. The generation of errors. We also tried to limit these threats by choosing datasets that cover a variety of applications. We also included widely used datasets such as Mnist and Fashion-mnist, which increases the reproductibility of our results. The classification models we chose represent a wide selection of different classification approaches. By surveying experts and using the weighted mean of their evaluations, we limit the external threat of the evaluation of the difficulty to do elementary tasks. Moreover, the elementary tasks were obtained by decomposing repairing methods from multiple papers [7,14,10,5,16,8,4,2,11,18,1,13]. For the generation of errors, we generated them randomly by means of a uniform distribution in datasets and repeated the process 30 times to reduce bias.

5 Conclusion

We investigated the question: Is it always better to repair data? And studied five criteria C1, C2, C3, C4, and C5. There is not a single common answer for all situations to our research question but we were able to answer it for specific situations and give elements to help answer it in other situations such as the difficulty to use a repairing method.

Extensions of this work to applications other than classification tasks are possible. Additional research could include more data types than numeric, especially more complex data types such as time series, which would imply more possible types of errors.

References

1. Abedjan, Z., Chu, X., Deng, D., Fernandez, R.C., Ilyas, I.F., Ouzzani, M., Papotti, P., Stonebraker, M., Tang, N.: Detecting data errors: Where are we and what needs to be done? Proceedings of the VLDB Endowment **9**(12), 993–1004 (2016)

2. Ataeyan, M., Daneshpour, N.: A novel data repairing approach based on constraints and ensemble learning. Expert Systems with Applications **159** (2020). https://doi.org/https://doi.org/10.1016/j.eswa.2020.113511
3. Blake, C., Merz, C.: Uci repository of machine learning databases (1998)
4. Chu, Xu, I.F.I., Papotti, P.: Holistic data cleaning: Putting violations into context. IEEE 29th International Conference on Data Engineering (ICDE) (2013)
5. Chu, X., Ilyas, I.F.: Qualitative data cleaning. Proceedings of the VLDB Endowment **9**(13), 1605–1608 (2016)
6. Chu, X., Ilyas, I.F., Krishnan, S., Wang, J.: Data cleaning: Overview and emerging challenges. SIGMOD Proceedings of the 2016 international conference on management of data p. 2201–2206 (2016). https://doi.org/http://dx.doi.org/10.1145/2882903.2912574
7. Chu, X., Morcos, J., Ilyas, I.F., Ouzzani, M., Papotti, P., Tang, N., Ye, Y.: Katara: A data cleaning system powered by knowledge bases and crowdsourcing. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data. pp. 1247–1261 (2015)
8. Hynes, N., Sculley, D., Terry, M.: The Data Linter: Lightweight, Automated Sanity Checking for ML Data Sets (2017)
9. Krishnan, S., Franklin, M.J., Goldberg, K., Wu, E.: Boostclean: Automated error detection and repair for machine learning. arXiv preprint arXiv:1711.01299 (2017)
10. Li, P., Rao, X., Blase, J., Zhang, Y., Chu, X., Zhang, C.: Cleanml: A study for evaluating the impact of data cleaning on ml classification tasks. 36th IEEE International Conference on Data Engineering (ICDE 2020)(virtual) (2021)
11. Li, Y., Vasconcelos, N.: Repair: Removing representation bias by dataset resampling. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition p. 9572–9581 (2019)
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
13. Qahtan, A., Tang, N., Ouzzani, M., Cao, Y., Stonebraker, M.: Pattern functional dependencies for data cleaning. Proceedings of the VLDB Endowment **13**(5), 684–697 (2020)
14. Rekatsinas, T., Chu, X., Ilyas, I.F., Ré, C.: Holoclean: Holistic data repairs with probabilistic inference. Proceedings of the VLDB Endowment **10**(11) (2017)
15. Roxane Jouseau, Sebastien Salva, C.S.: Technical report for the paper: On studying the effect of data quality on classification performance. https://gitlab.com/roxane.jouseau/is-it-always-better-to-repair-data (2022)
16. Wu, R., Chaba, S., Sawlani, S., Chu, X., Thirumuruganathan, S.: Zeroer: Entity resolution using zero labeled examples. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. pp. 1149–1164 (2020)
17. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)
18. Zhang, X., Ji, Y., Nguyen, C., Wang, T.: Deepclean: Data cleaning via question asking. 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA) p. 283–292 (2018)