Technical report for: On Studying the Effect of Data Quality on Classification Performance

roxane jouseau, Sébastien Salva, Chafik Samir

Technical report for: On Studying the Effect of Data Quality on Classification Performance

Abstract
This document presents additional experimental results and details that could not be included in the original paper because of space limitations.

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 [1, 2, 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

Table 1: Datasets

Name	Description	Dimension	Number of classes	Number of entries
Mnist [1]	8x8 grayscale images	64	10	1797
Fashion mnist [2]	28x28p grayscale images	784	10	700
Olivetti [1]	64x64 grayscale face images taken between April 1992 and April 1994 at AT&T Laboratories Cambridge	4096	40	400
Iris [1]	measures of 3 different types of irises	4	3	150
Adult [3]	income data from the census	10	2	2000
Breast cancer [1]	characteristics of a cell nuclei present in an image to classify between benign or malignant	13	2	569
Wine [1]	results of a chemical analysis of wines grown in the same region in Italy by 3 different cultivators	13	3	178

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 1.

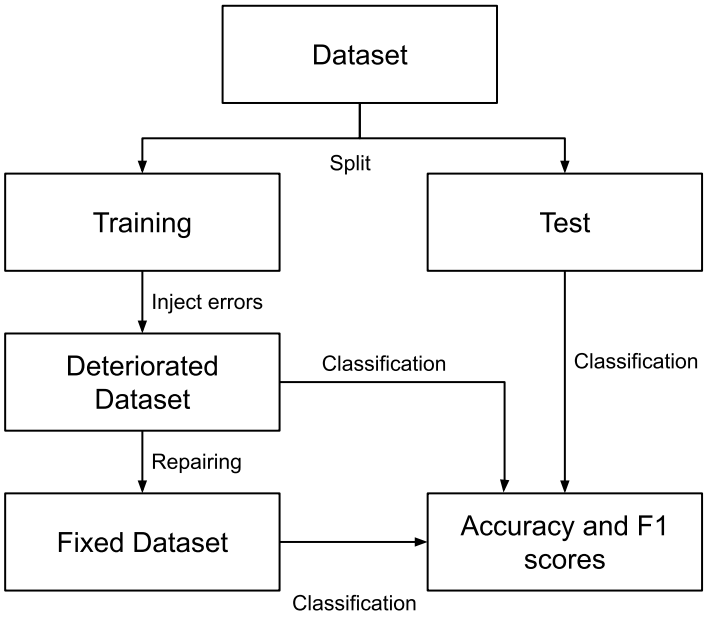
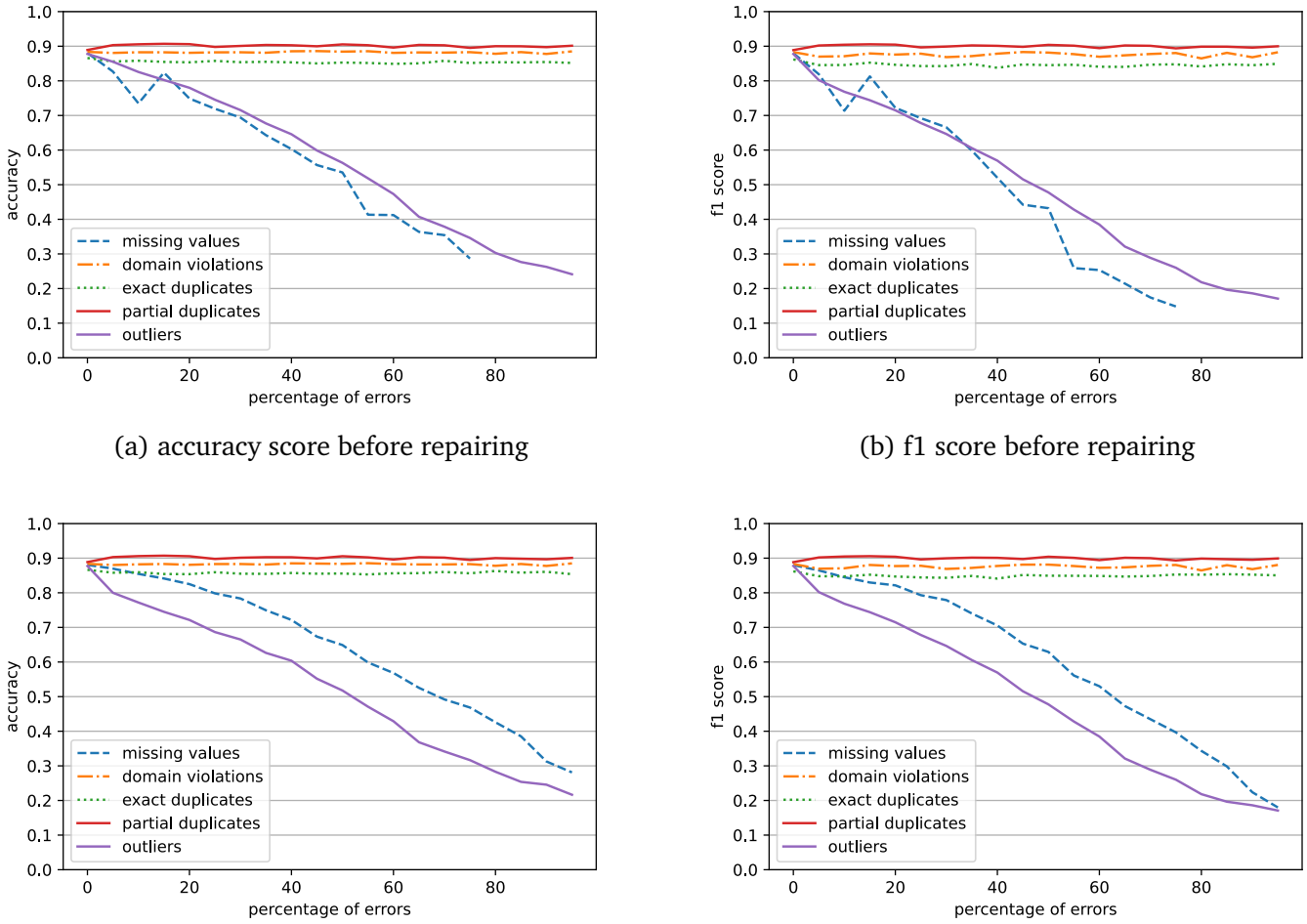


Figure 1: Structure of the experiment

2 Experimental results

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

In order to observe the impact of the degradation of data on repairing effectiveness 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 2a and 2b, we respectively depict the mean accuracies and F1 scores of the classification models on all the datasets by error type as a function of the percentage of errors injected in the data. Figure 2c and 2d respectively display the mean accuracies and f1 scores of the classification models on all the datasets after repair as a function of the percentage of errors injected in the data.



(a) accuracy score before repairing

(b) f1 score before repairing

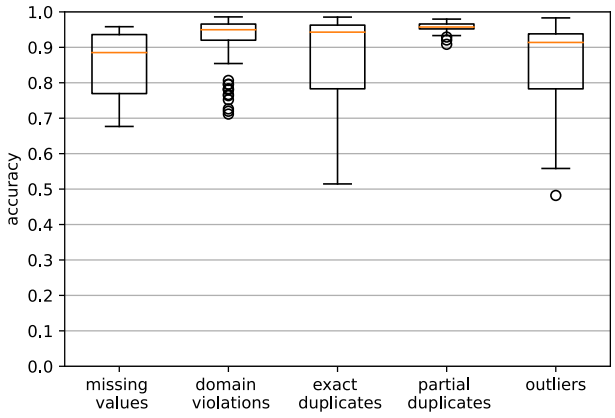
(c) accuracy score after repairing

(d) f1 score after repairing

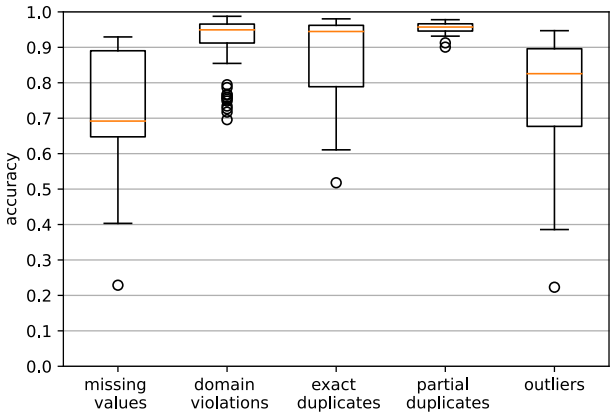
Figure 2: Impact of the degradation of the data on repairing effectiveness

2.2 C3: impact of the type of error

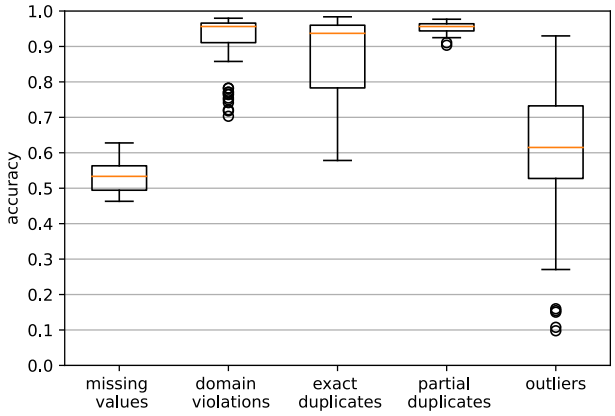
In order to study the impact of the type of errors present in data we randomly injected different types of errors in data with different levels of errors. We then trained classification models on data before and after repairing. Figure 3 boxplots the accuracy for each error type before repairing at a given level of deterioration (10%, 25%, 50%, and 80%) and Figure 4 boxplot the accuracy for the different error types at the same levels of deterioration after repairing.



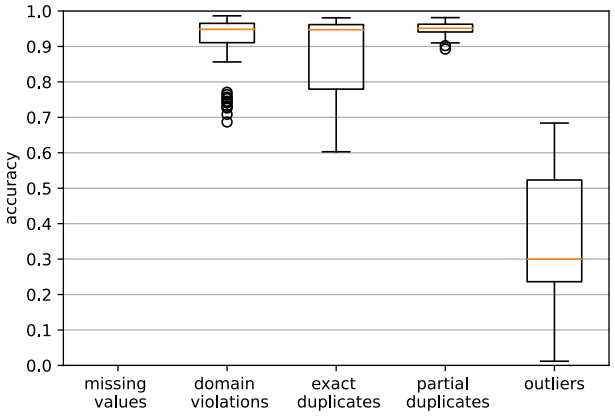
(a) accuracy for 10% of errors before repairing



(b) accuracy for 25% of errors before repairing

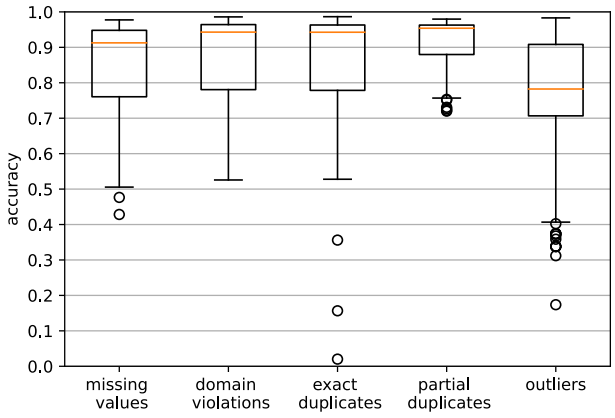


(c) accuracy for 50% of errors before repairing

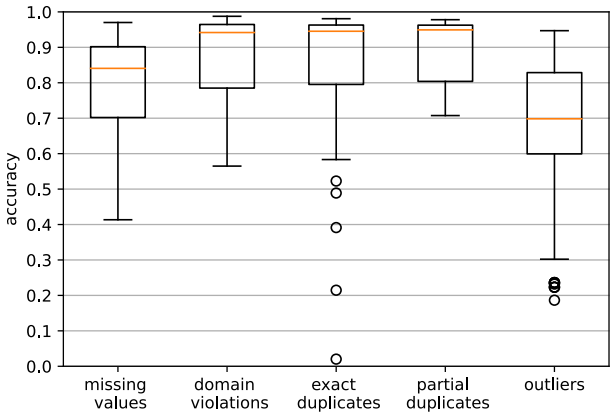


(d) accuracy for 80% of errors before repairing

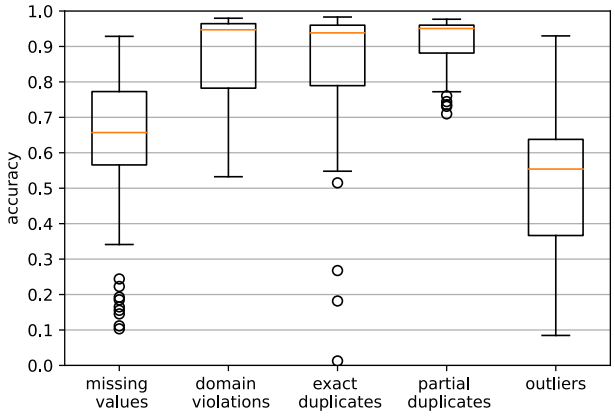
Figure 3: Accuracy by error type without repairing



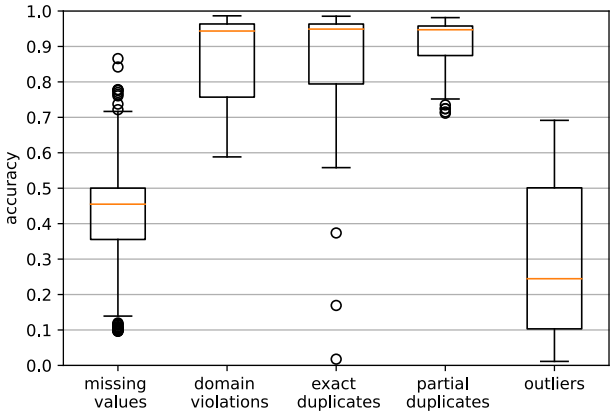
(a) accuracy for 10% of errors after repairing



(b) accuracy for 25% of errors after repairing



(c) accuracy for 50% of errors after repairing

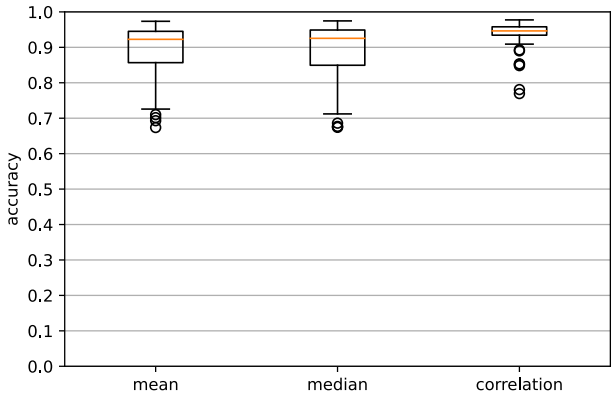


(d) accuracy for 80% of errors after repairing

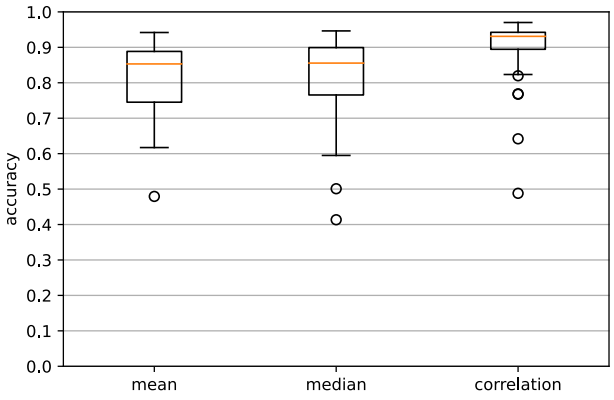
Figure 4: Accuracy by error type after repairing

2.3 C4: effectiveness of the repairing tool

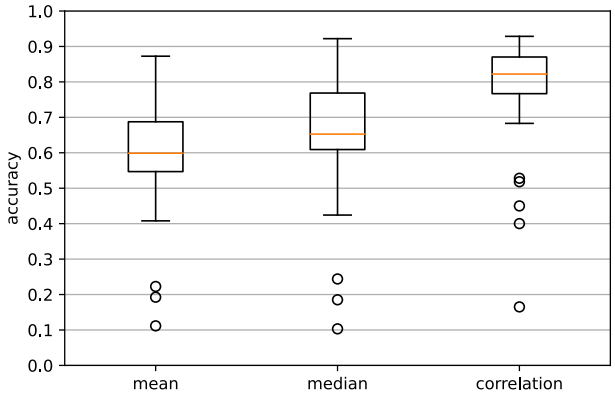
To study the effectiveness of repairing tools we randomly injected errors at increasing percentages from 0 to 95% in data. We then repaired datasets with different repairing methods, trained the classification models on those datasets and retrieved their accuracies and f1-scores. We only show the repairing of missing values and outliers as their degradation has the most impact compared to the their types of errors. Figure 5 shows the accuracy for 10%, 25%, 50%, and 75% of missing values after repairing with the methods R_med, R_mean, and R_correl. Figure 6 depicts the accuracy for 10%, 25%, 40%, and 50% of outliers after repairing with the methods R_std, R_quart, R_quant, and R_linter.



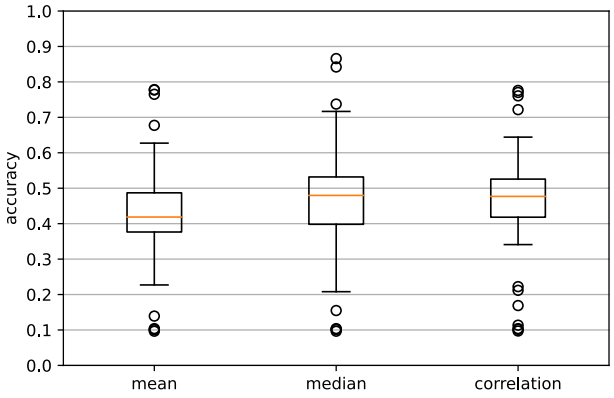
(a) accuracy for 10% of errors



(b) accuracy for 25% of errors

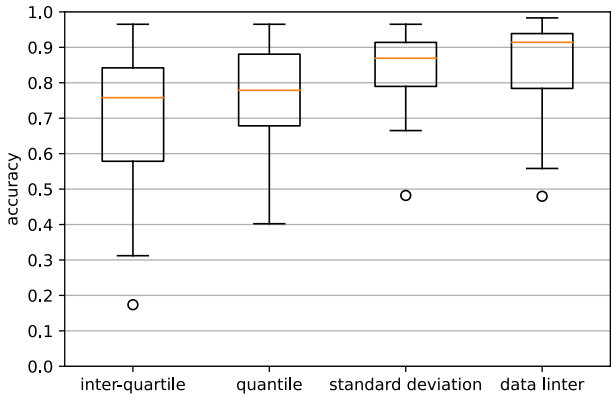


(c) accuracy for 50% of errors

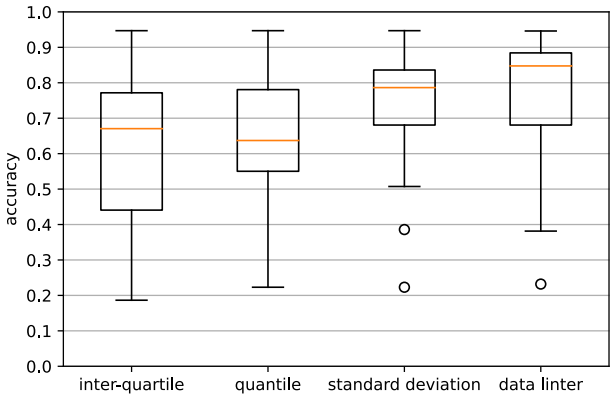


(d) accuracy for 75% of errors

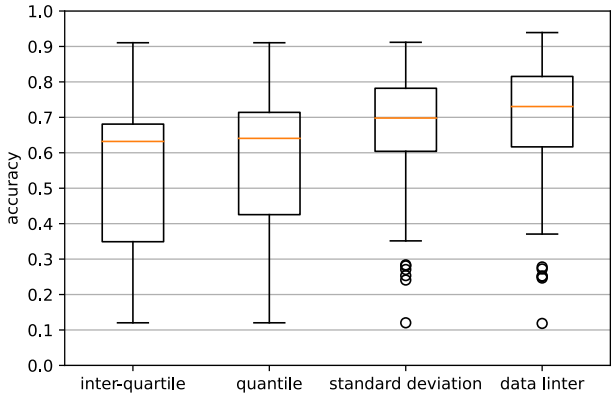
Figure 5: effectiveness of some repairing tools on missing values



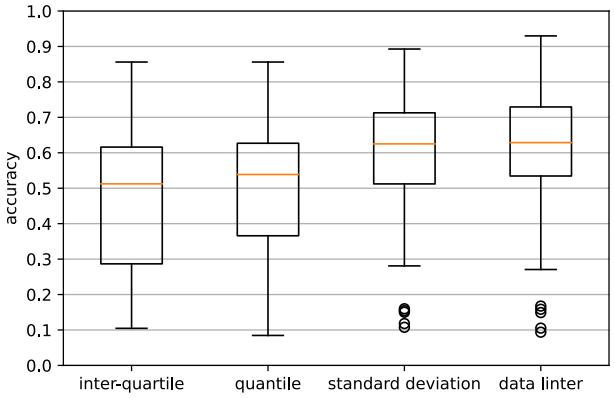
(a) accuracy for 10% of errors



(b) accuracy for 25% of errors



(c) accuracy for 40% of errors

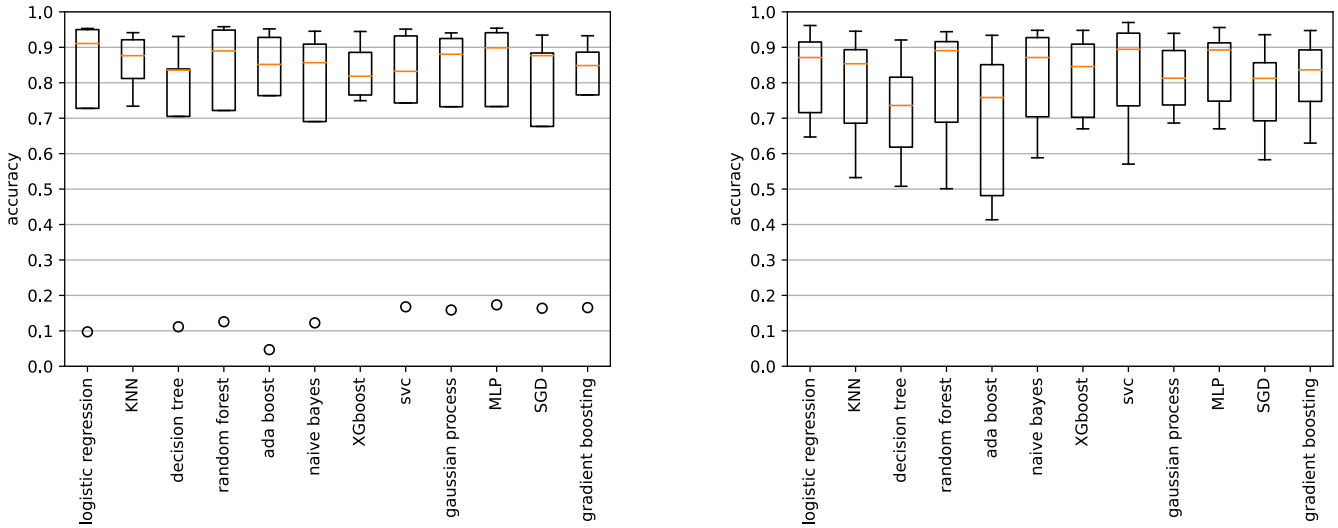


(d) accuracy for 50% of errors

Figure 6: effectiveness of some repairing tools on outliers

2.4 C5: impact of the classification model

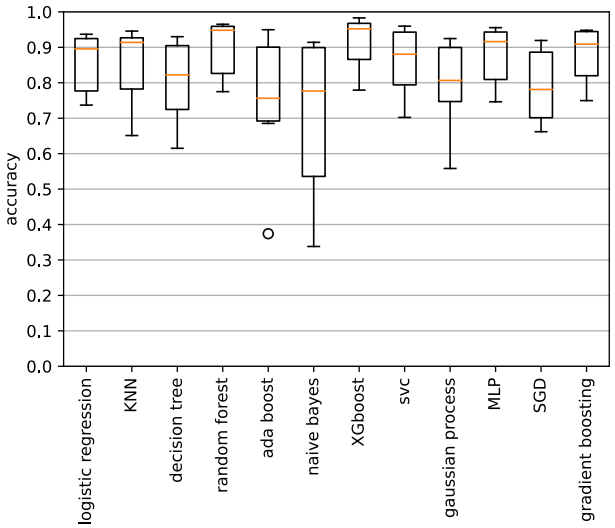
In order to study the impact of the classification model we observe the accuracy score for each classification model for a fixed percentage of missing values or outliers injected in data. Figure 7 depicts the accuracy score for missing values and Figure 8 for outliers at fixed levels of degradation for the different classification models before repairing. We only display missing values for 10% and 25% of degradation here as for higher levels of degradation (50%, 80%) the high number of missing values causes classes to disappear, we then don't have enough data to analyze. Figure 9 and Figure 10 respectively depict the accuracy score for missing values and outliers at 10%, 25%, 50%, and 80% of errors for the different classification models after repairing.



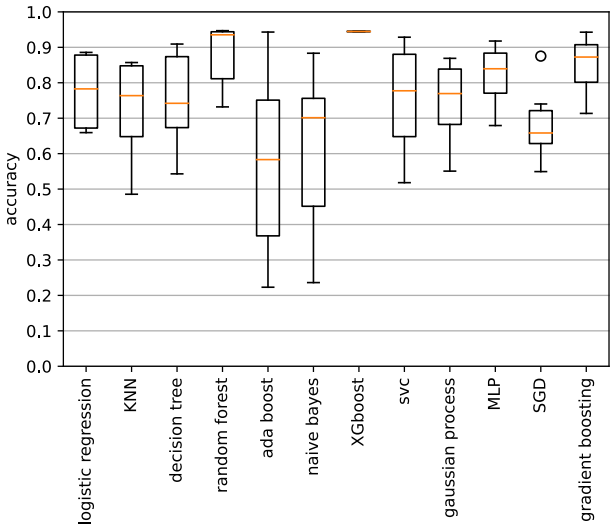
(a) accuracy for 10% of missing values

(b) accuracy for 25% of missing values

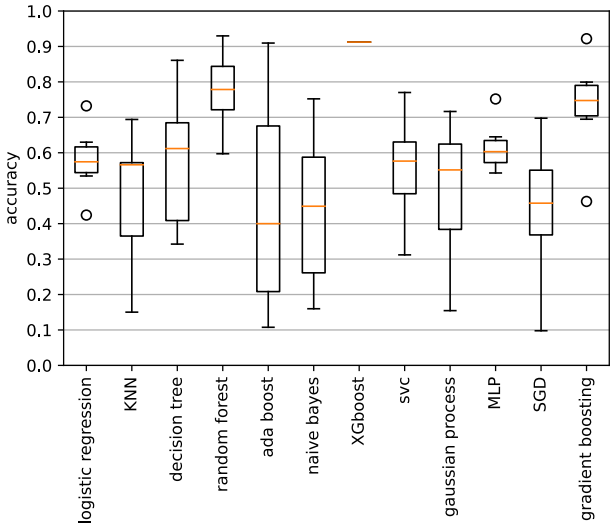
Figure 7: accuracy for different classification models at different levels of missing values



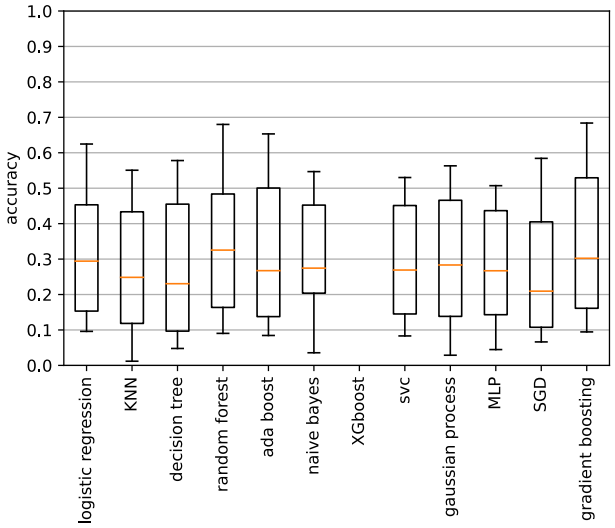
(a) accuracy for 10% of outliers



(b) accuracy for 25% of outliers

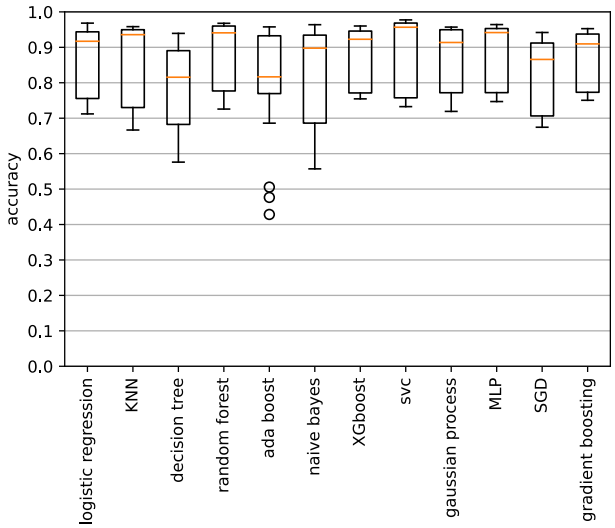


(c) accuracy for 50% of outliers

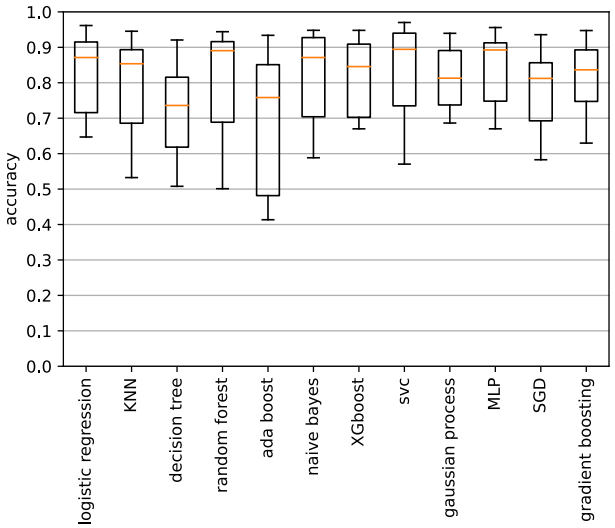


(d) accuracy for 80% of outliers

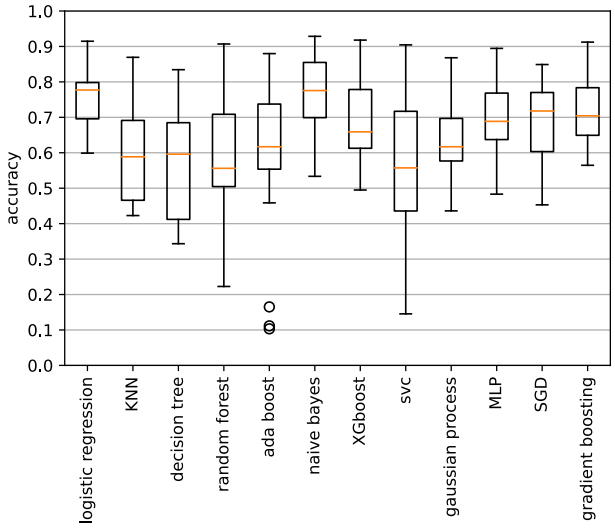
Figure 8: accuracy for different classification models at different levels of outliers before repairing



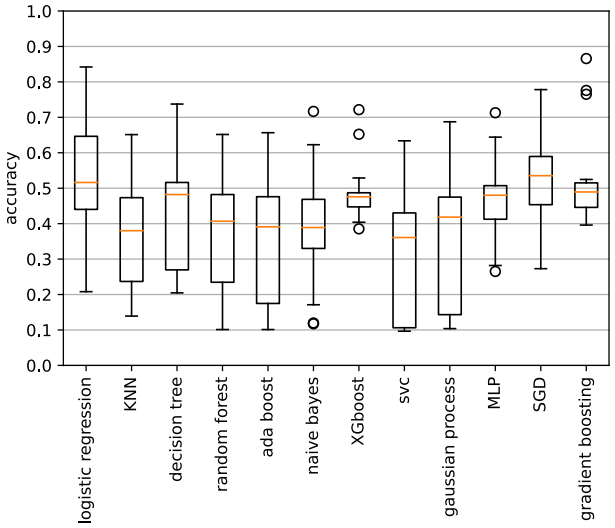
(a) accuracy for 10% of missing values after repairing



(b) accuracy for 25% of missing values after repairing

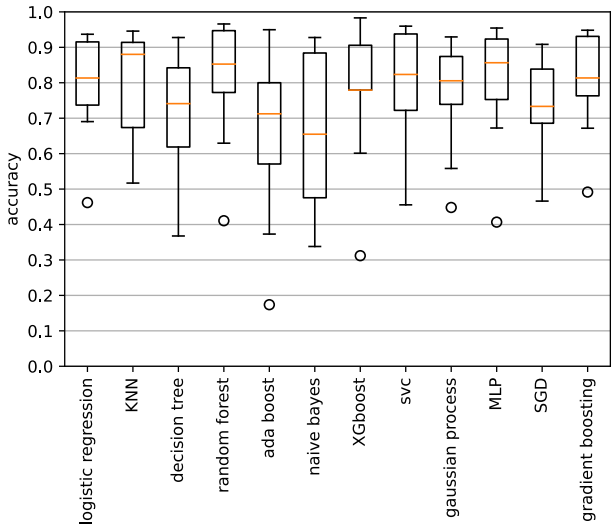


(c) accuracy for 50% of missing values after repairing

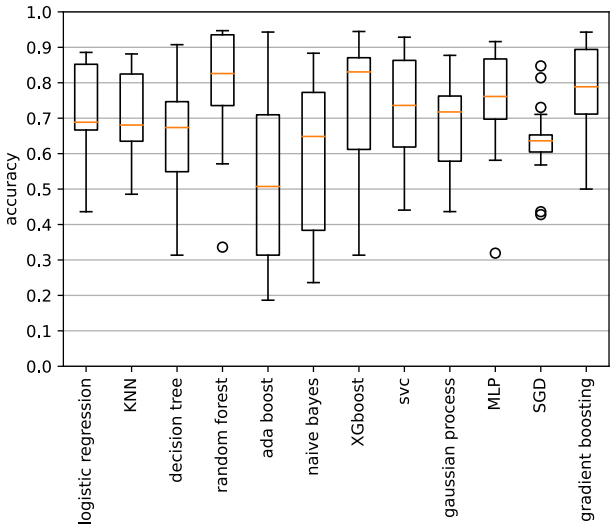


(d) accuracy for 80% of missing values after repairing

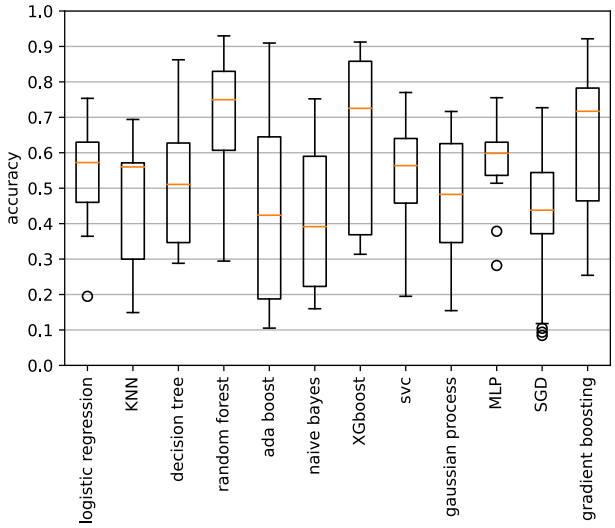
Figure 9: accuracy for different classification models at different levels of missing values after repairing



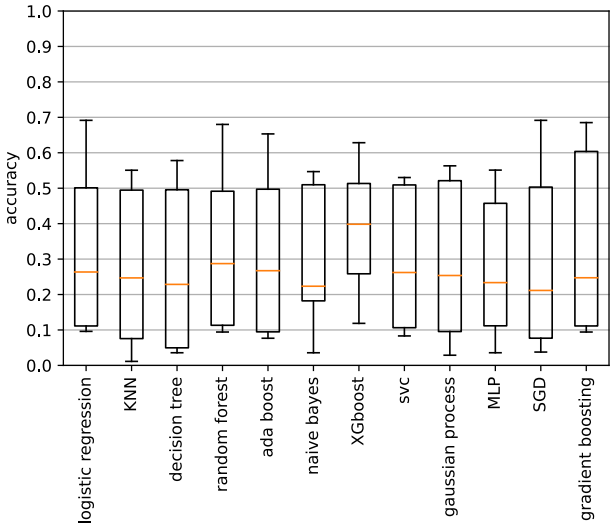
(a) accuracy for 10% of outliers after repairing



(b) accuracy for 25% of outliers after repairing



(c) accuracy for 50% of outliers after repairing



(d) accuracy for 80% of outliers after repairing

Figure 10: accuracy for different classification models at different levels of outliers after repairing

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[2] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[3] C. Blake and C. Merz. Uci repository of machine learning databases, 1998.