CleanML Study: Investigating the Impact of Data Cleaning on ML Classification Tasks

A Comprehensive Analysis of 14 Real-World Datasets

By Kayvan Shah

Background

Overview

- Data quality significantly impacts ML model performance, leading to data scientists spending substantial time on data cleaning before training models.
- Currently, there is a lack of rigorous studies on how data cleaning specifically influences ML. ML focuses on robust algorithms against certain noise types, while the database community primarily studies data cleaning without considering downstream ML analytics.
- CleanML includes 14 real-world datasets with real errors, covers five common error types, involves seven ML models, and utilizes various cleaning algorithms, including commonly used and state-of-the-art solutions.
- The research introduces CleanML, aiming to systematically investigate the impact of data cleaning on ML classification tasks.

Introduction

- The quality of ML applications relies on data quality, and data cleaning is essential for building high-quality ML models.
- Both ML and database communities address issues related to dirty data, but there is a disconnect in their approaches. ML community focuses on understanding noise impact on models without actual data cleaning, while the DB community focuses on data cleaning processes without considering ML impacts.
- The goal is to conduct the first systematic empirical study on the impact of data cleaning on downstream ML models.

Related Work - Navigating the Landscape of Data Cleaning

From Traditional Techniques to Advanced ML Approaches

- Recent studies suggest that even advanced error detection techniques may miss errors in real-world datasets.
- Analytics-driven cleaning methods aim to address the challenges of expensive and hard-to-reach ground truth in data cleaning.
- CleanML, in contrast, evaluates automatic cleaning algorithms, considering five error types and seven ML models.

CleanML vs Other Cleaning Methods

- SampleClean focuses on answering SQL aggregate queries with statistical guarantees, while ActiveClean cleans data for convex ML models trained using gradient descent.
- BoostClean selects cleaning methods using statistical boosting but only considers domain value violations and a specific ML model.
- CleanML, in contrast, evaluates automatic cleaning algorithms, considering five error types and seven ML models.

CleanML Database Schema & Experimental Design

CleanML Schema & Relations

R1 (Vanilla)						
Dataset	Error Type	Detection	Repair	ML Model	Scenario	Flag
R2 (With Model Selection)						
Dataset	Error Type	Detection	Repair	Scenario	Flag	
R3 (With Model Selection and Cleaning Method Selection)						
Dataset	Error Type	Scenario	Flag			

Three Relations:

- R1 (Vanilla): Original version with dataset, error type, detection, repair, ML model, and scenario.
- R2 (With Model Selection): Eliminates ML model attribute, focuses on model selection.
- R3 (With Model Selection and Cleaning Method Selection): Further eliminates cleaning method attributes, focuses on both model and cleaning method selection.

Attributes in CleanML Relations

- Dataset Attribute:
  - Specifies the dataset under study, associated with an ML task.
  - Utilizes real-world datasets with various errors rather than synthetic errors.
- Attributes for Data Cleaning:
  - Includes error type, considering missing values, outliers, duplicates, inconsistencies, and mislabels.
  - Considers commonly used cleaning methods for each error type.
  - Examines single error cleaning as well as mixed error cleaning.
- Attribute for ML Model:
  - Specifies the ML algorithm used for classification.
  - Different ML models may exhibit varying behaviors to different error types.
- Attribute for Cleaning Scenario:
  - Indicates whether cleaning is applied in the training or test set.
  - Determines whether the evaluation focuses on model development or deployment.
- Flag Attribute:
  - Summarizes the impact of data cleaning on ML.
  - Values: "P (positive)," "N (negative)," or "S (insignificant)."

Experimental Design

- Error Types and Automatic Cleaning Methods:
    - Error Types: Missing values, outliers, duplicates, inconsistencies, mislabels.
    - Automatic Cleaning Methods: Various methods for each error type, including deletion, imputation, HoloClean, and others.
- Datasets:
    - 14 real-world datasets with varying error types and rates, including inconsistencies, duplicates, missing values, outliers, and mislabels.
- ML Models:
    - Seven classical and competitive ML models used in classification tasks, including Logistic Regression, KNN, Decision Tree, Random Forest, AdaBoost, Naive Bayes, and XGBoost.

Experimental Design

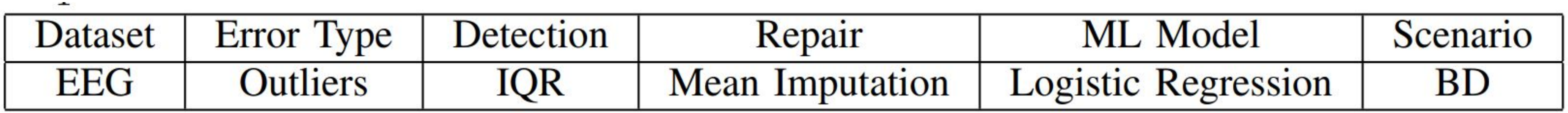
- Scenarios:
  - Data cleaning applied in the model development or deployment phase.
  - Four performance metrics based on different scenarios: A, B, C, D.
  - Special consideration for missing values, treating deletion and imputation separately.
- Special Consideration for Missing Values:
  - Different handling for missing values due to the inability to train or predict on datasets with missing values.
  - Model development scenario (BD) for missing values, treating deletion as the "dirty" dataset and cleaning by filling missing values.

Scenarios

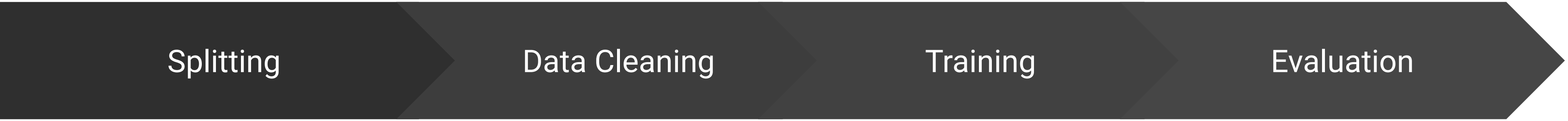
Training\Test	**Dirty**	**Clean**
Dirty	A	B
Clean	C	D

Training\Test	**Deletion**	**Imputation**
Deletion	A	B
Imputation	C	D

CleanML Database Instance



Dataset	Error Type	Detection	Repair	ML Model	Scenario
EEG	Outliers	IQR	Mean Imputation	Logistic Regression	BD



Splitting

Data Cleaning

Training

Evaluation

Train and test in 80/20 ratio

Cleaning of error type from train and test set is done based upon the cleaning algorithm mentioned in the table
To avoid any data leakage, all statistics necessary are computed on training set and used on both

Number of models trained depends on the scenario

Same number of models are evaluated on clean and dirty set to get a list of performance metrics

Generating one instance

Handling Randomness & Significant Testing

Addressing ML Randomness:

- ML experiments involve randomness, resulting in different metric pairs for various train/test splits.
- To mitigate this, the procedure is conducted 20 times with different splits, providing a more robust assessment.

Statistical Significance Testing:

- Averaging metric pairs may not capture the significance of the difference between metrics effectively.
- Proposed approach involves rigorous statistical significance testing using the paired sample t-test.

Paired Sample t-Test Approach:

- Utilizes the paired sample t-test, commonly employed to determine mean differences in observations.
- Tests include two-tailed, upper-tailed, and lower-tailed variations to comprehensively assess significance.

Three-Valued Flag Attribute:

- The challenge lies in producing a three-valued flag attribute based on the three tests.
- Flag assignment is determined considering p-values and significance levels.

Handling Randomness & Significant Testing

Mitigating Test Result Ambiguity:

- The challenge of symmetric test statistics distribution is addressed by conducting three tests and selectively reporting results.
- Reporting the one-tailed test results only if the two-tailed test is significant, preventing ambiguity in result interpretation.

Flag Determination Process:

- If $p0 \geq \propto$, the flag is set to "S" (insignificant).
- If $p0 < \propto$ and both $p1 < \propto$ and $p2 < \propto$, the flag is set to "P" (positive).
- If $p0 < \propto$ and either p1 or $p2 < \propto$, the flag is set to "N" (negative).

p0, p1, p2 denote the p-values of two-tailed t-test, upper-tailed t-test, and lower-tailed t-test respectively

$\propto$ denotes the significant level

Controlling False Discoveries

- Scale of Multiple Hypothesis Testing:
  - Challenge: The study involves testing numerous hypotheses, leading to a common statistical issue—false discoveries.
  - Magnitude: With 3612, 516, and 168 hypotheses in R1, R2, and R3, respectively, the likelihood of false discoveries is substantial.
- Approaches to Control False Discovery Rate (FDR):
  - Traditional Method: Bonferroni correction adjusts threshold for each test but may falter when faced with a greater number of non-significant results.
  - FDR Strategy: Ranks tests by p-values and selects a set number of top-ranked tests as significant.
  - Common FDR Procedures: Implementation of Benjamini-Yekutieli (BY) procedure, which controls FDR under arbitrary dependence assumptions.
- Rigorous FDR Control in the Study:
  - Separate Procedures: A distinct BY procedure is conducted for each relation (R1, R2, R3) to effectively manage false discovery rate.
  - Threshold = 0.05: Ensures stringent control over significance declarations, maintaining the reliability of the study's findings.

Result Analysis on CleanML Database

Strategy Used

SQL Query Templates:

- Query templates are employed on relations R1, R2, and R3 (CleanML database) for result analysis.
- Error types (E) considered: inconsistencies, duplicates, mislabels, outliers, missing values.

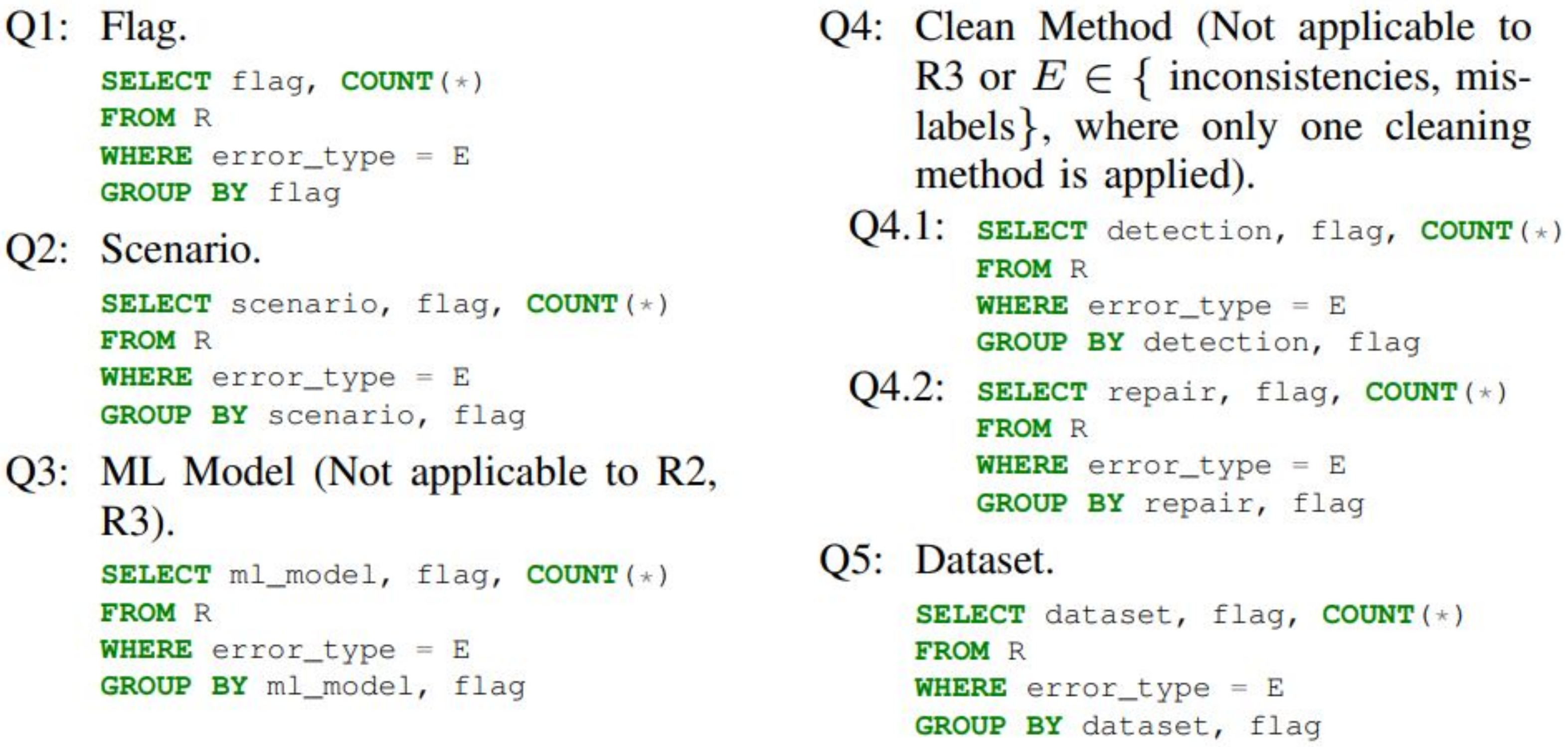
Comparison between R1, R2, and R3:

- Assesses the differences in results obtained from the same query template issued against R1, R2, and R3.
- Explores whether model selection (R2) and cleaning algorithm selection (R3) contribute to a positive impact on data cleaning for downstream ML models.

Varying Granularity of Analysis:

- Q1: Aggregates over all datasets, ML models, scenarios, and cleaning methods, fixing the error type, to assess the general impact of cleaning.
- Q2: Groups by an additional attribute (e.g., scenario, ML model) to identify scenarios where the impact differs from the general trend observed in Q1.
- Q3: Analyzes the impact based on ML models.
- Q4.1 and Q4.2: Examines the impact based on cleaning methods (detection and repair).
- Q5: Investigates the impact across different datasets.

SQL Queries used for Analysis



Q1: Flag.

```
SELECT flag, COUNT(*)
FROM R
WHERE error_type = E
GROUP BY flag
```

Q2: Scenario.

```
SELECT scenario, flag, COUNT(*)
FROM R
WHERE error_type = E
GROUP BY scenario, flag
```

Q3: ML Model (Not applicable to R2, R3).

```
SELECT ml_model, flag, COUNT(*)
FROM R
WHERE error_type = E
GROUP BY ml_model, flag
```

Q4: Clean Method (Not applicable to R3 or $E \in \{$ inconsistencies, mislabels$\}$, where only one cleaning method is applied).

Q4.1:

```
SELECT detection, flag, COUNT(*)
FROM R
WHERE error_type = E
GROUP BY detection, flag
```

Q4.2:

```
SELECT repair, flag, COUNT(*)
FROM R
WHERE error_type = E
GROUP BY repair, flag
```

Q5: Dataset.

```
SELECT dataset, flag, COUNT(*)
FROM R
WHERE error_type = E
GROUP BY dataset, flag
```

Missing Values

- Cleaning missing values by imputation is more likely to improve or achieve similar ML performance compared to deletion.
- Model and imputation method selection increase the likelihood of positive impacts, reducing negative impacts.
- Impact varies significantly across different datasets.
- Advanced cleaning methods, such as HoloClean, do not notably outperform simple imputation methods.

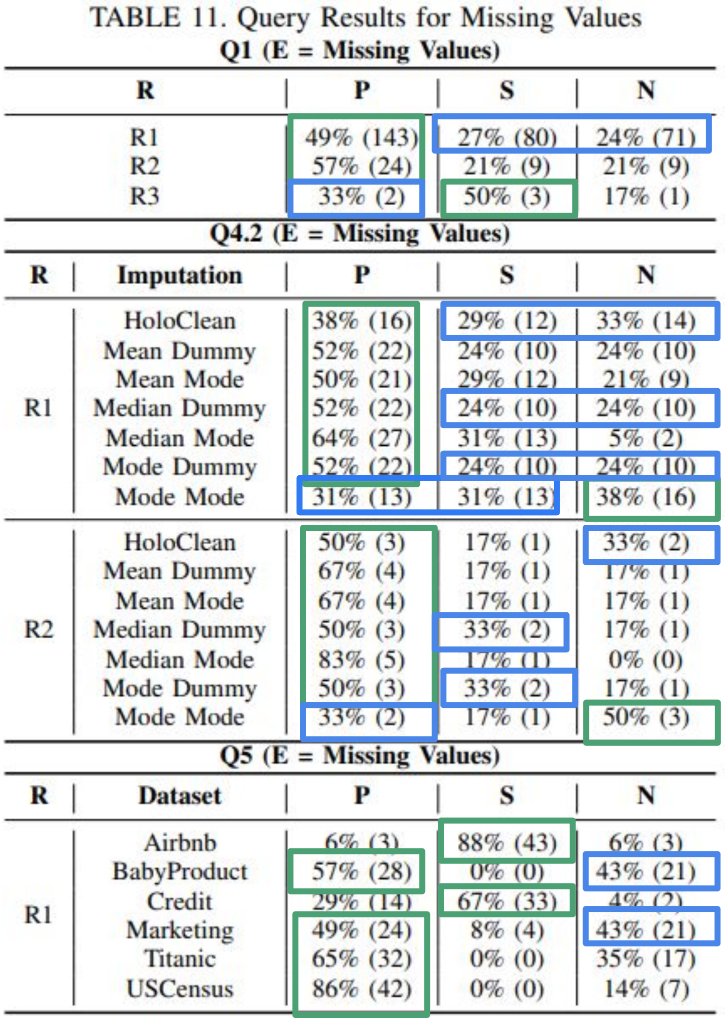


TABLE 11. Query Results for Missing Values

Q1 (E = Missing Values)

R	P	S	N
R1	49% (143)	27% (80)	24% (71)
R2	57% (24)	21% (9)	21% (9)
R3	33% (2)	50% (3)	17% (1)

Q4.2 (E = Missing Values)

R	Imputation	P	S	N
R1	HoloClean	38% (16)	29% (12)	33% (14)
	Mean Dummy	52% (22)	24% (10)	24% (10)
	Mean Mode	50% (21)	29% (12)	21% (9)
	Median Dummy	52% (22)	24% (10)	24% (10)
	Median Mode	64% (27)	31% (13)	5% (2)
	Mode Dummy	52% (22)	24% (10)	24% (10)
	Mode Mode	31% (13)	31% (13)	38% (16)
R2	HoloClean	50% (3)	17% (1)	33% (2)
	Mean Dummy	67% (4)	17% (1)	17% (1)
	Mean Mode	67% (4)	17% (1)	17% (1)
	Median Dummy	50% (3)	33% (2)	17% (1)
	Median Mode	83% (5)	17% (1)	0% (0)
	Mode Dummy	50% (3)	33% (2)	17% (1)
	Mode Mode	33% (2)	17% (1)	50% (3)

Q5 (E = Missing Values)

R	Dataset	P	S	N
R1	Airbnb	6% (3)	88% (43)	6% (3)
	BabyProduct	57% (28)	0% (0)	43% (21)
	Credit	29% (14)	67% (33)	4% (2)
	Marketing	49% (24)	8% (4)	43% (21)
	Titanic	65% (32)	0% (0)	35% (17)
	USCensus	86% (42)	0% (0)	14% (7)

Outliers

- Cleaning outliers is more likely to have an insignificant impact on model performance.
- Model and cleaning algorithm selection can significantly reduce the probability of negative impacts.
- Impact varies vastly across datasets.
- Different detection methods have major differences in observed impact.

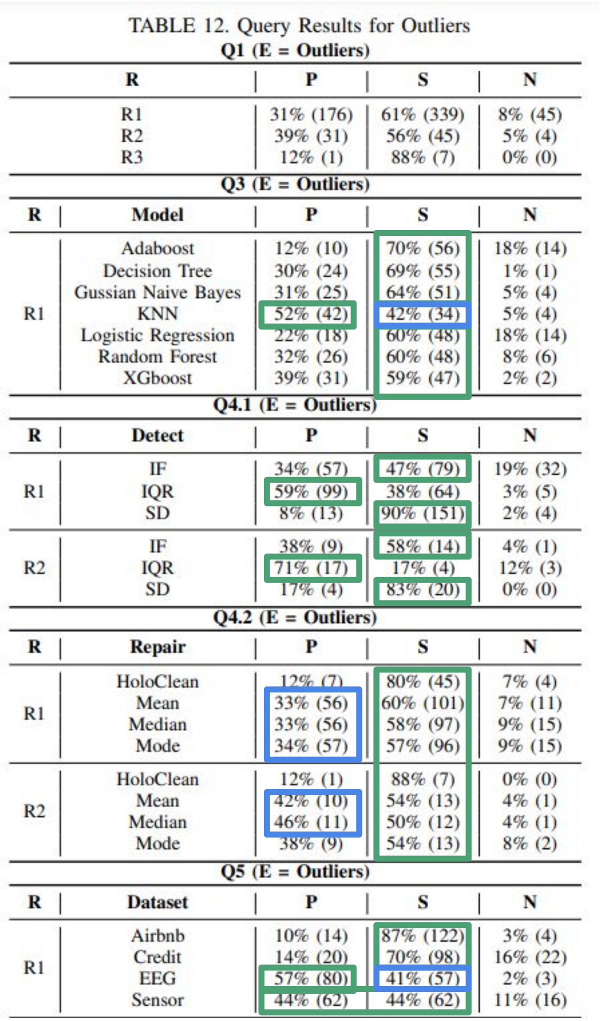


TABLE 12. Query Results for Outliers

Q1 (E = Outliers)

R	P	S	N
R1	31% (176)	61% (339)	8% (45)
R2	39% (31)	56% (45)	5% (4)
R3	12% (1)	88% (7)	0% (0)

Q3 (E = Outliers)

R	Model	P	S	N
R1	Adaboost	12% (10)	70% (56)	18% (14)
	Decision Tree	30% (24)	69% (55)	1% (1)
	Gussian Naive Bayes	31% (25)	64% (51)	5% (4)
	KNN	52% (42)	42% (34)	5% (4)
	Logistic Regression	22% (18)	60% (48)	18% (14)
	Random Forest	32% (26)	60% (48)	8% (6)
	XGboost	39% (31)	59% (47)	2% (2)

Q4.1 (E = Outliers)

R	Detect	P	S	N
R1	IF	34% (57)	47% (79)	19% (32)
	IQR	59% (99)	38% (64)	3% (5)
	SD	8% (13)	90% (151)	2% (4)
R2	IF	38% (9)	58% (14)	4% (1)
	IQR	71% (17)	17% (4)	12% (3)
	SD	17% (4)	83% (20)	0% (0)

Q4.2 (E = Outliers)

R	Repair	P	S	N
R1	HoloClean	12% (7)	80% (45)	7% (4)
	Mean	33% (56)	60% (101)	7% (11)
	Median	33% (56)	58% (97)	9% (15)
	Mode	34% (57)	57% (96)	9% (15)
R2	HoloClean	12% (1)	88% (7)	0% (0)
	Mean	42% (10)	54% (13)	4% (1)
	Median	46% (11)	50% (12)	4% (1)
	Mode	38% (9)	54% (13)	8% (2)

Q5 (E = Outliers)

R	Dataset	P	S	N
R1	Airbnb	10% (14)	87% (122)	3% (4)
	Credit	14% (20)	70% (98)	16% (22)
	EEG	57% (80)	41% (57)	2% (3)
	Sensor	44% (62)	44% (62)	11% (16)

Mislabels

- Cleaning mislabels is likely to have positive or insignificant impacts on ML.
- Model selection increases the likelihood of positive impacts.
- Boosting-based ML models are most reactive to mislabels.
- The impact varies significantly across datasets.

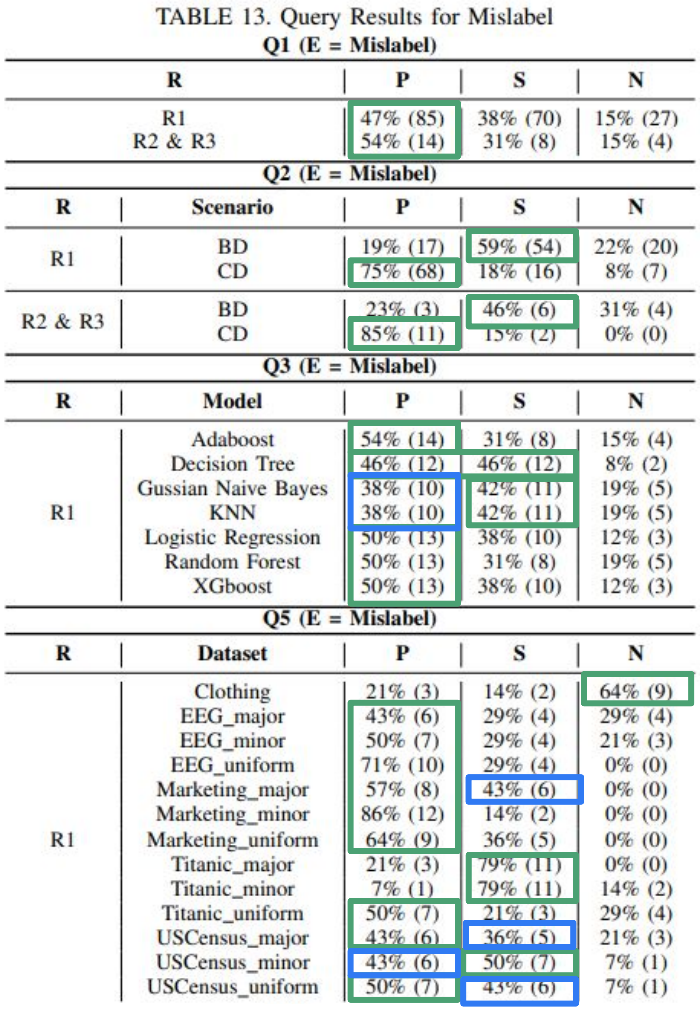


TABLE 13. Query Results for Mislabel

Q1 (E = Mislabel)

R	P	S	N
R1	47% (85)	38% (70)	15% (27)
R2 & R3	54% (14)	31% (8)	15% (4)

Q2 (E = Mislabel)

R	Scenario	P	S	N
R1	BD	19% (17)	59% (54)	22% (20)
	CD	75% (68)	18% (16)	8% (7)
R2 & R3	BD	23% (3)	46% (6)	31% (4)
	CD	85% (11)	15% (2)	0% (0)

Q3 (E = Mislabel)

R	Model	P	S	N
R1	Adaboost	54% (14)	31% (8)	15% (4)
	Decision Tree	46% (12)	46% (12)	8% (2)
	Gussian Naive Bayes	38% (10)	42% (11)	19% (5)
	KNN	38% (10)	42% (11)	19% (5)
	Logistic Regression	50% (13)	38% (10)	12% (3)
	Random Forest	50% (13)	31% (8)	19% (5)
	XGboost	50% (13)	38% (10)	12% (3)

Q5 (E = Mislabel)

R	Dataset	P	S	N
R1	Clothing	21% (3)	14% (2)	64% (9)
	EEG_major	43% (6)	29% (4)	29% (4)
	EEG_minor	50% (7)	29% (4)	21% (3)
	EEG_uniform	71% (10)	29% (4)	0% (0)
	Marketing_major	57% (8)	43% (6)	0% (0)
	Marketing_minor	86% (12)	14% (2)	0% (0)
	Marketing_uniform	64% (9)	36% (5)	0% (0)
	Titanic_major	21% (3)	79% (11)	0% (0)
	Titanic_minor	7% (1)	79% (11)	14% (2)
	Titanic_uniform	50% (7)	21% (3)	29% (4)
	USCensus_major	43% (6)	36% (5)	21% (3)
	USCensus_minor	43% (6)	50% (7)	7% (1)
	USCensus_uniform	50% (7)	43% (6)	7% (1)

Inconsistencies

- Cleaning inconsistencies is more likely to have an insignificant impact and is unlikely to have a negative impact on ML.
- The impact varies significantly across datasets.
- Selecting the best ML model increases the likelihood of positive impact after cleaning inconsistencies.

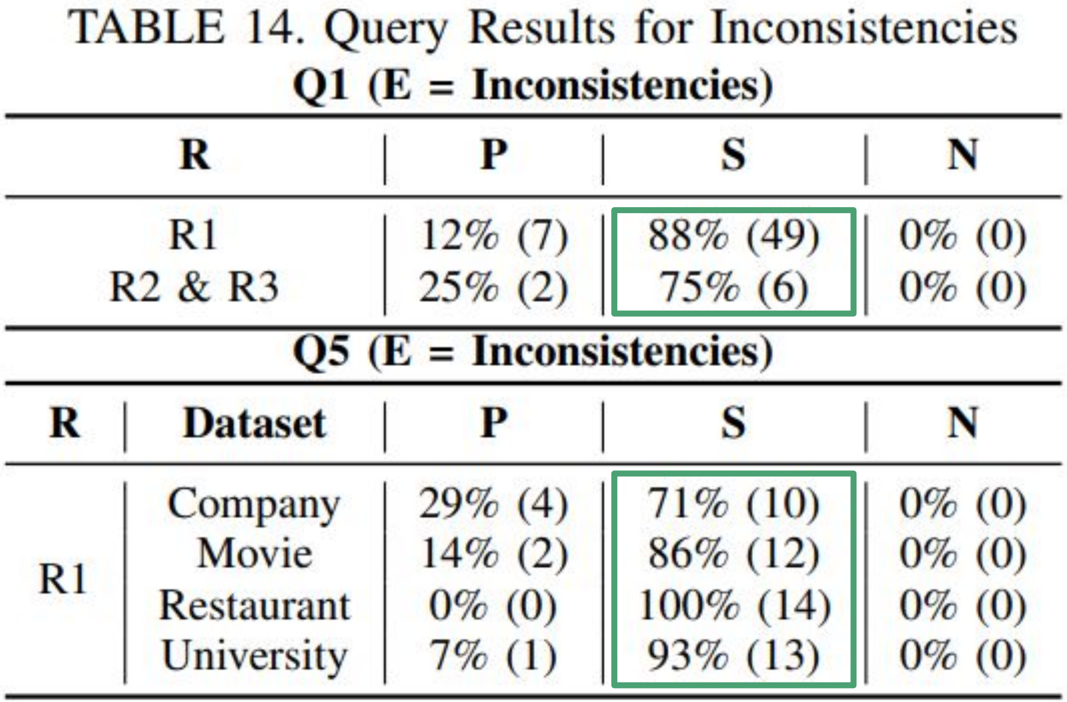


TABLE 14. Query Results for Inconsistencies

Q1 (E = Inconsistencies)

R	P	S	N
R1	12% (7)	88% (49)	0% (0)
R2 & R3	25% (2)	75% (6)	0% (0)

Q5 (E = Inconsistencies)

R	Dataset	P	S	N
R1	Company	29% (4)	71% (10)	0% (0)
	Movie	14% (2)	86% (12)	0% (0)
	Restaurant	0% (0)	100% (14)	0% (0)
	University	7% (1)	93% (13)	0% (0)

Duplicates

- Cleaning duplicates is more likely to have insignificant or negative impacts than positive impacts.
- The impact on cleaning duplicates varies across detection methods and datasets.

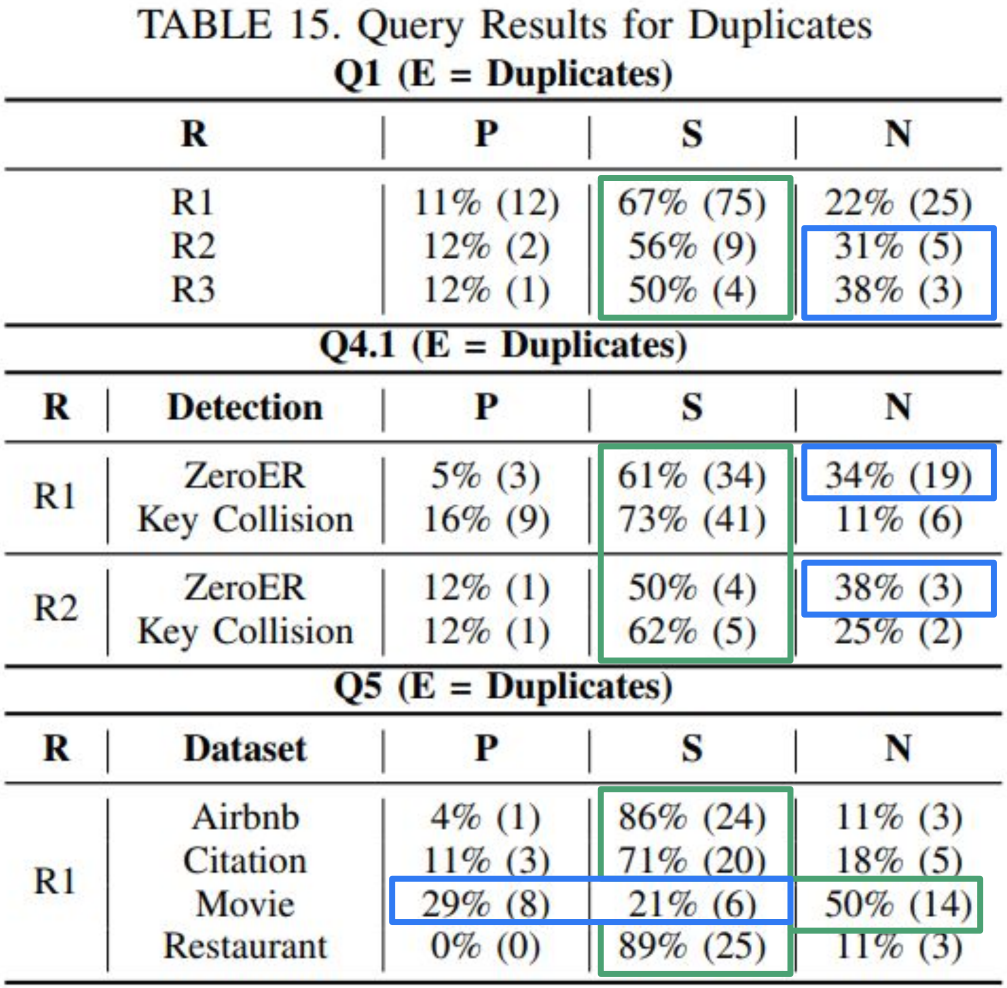


TABLE 15. Query Results for Duplicates

Q1 (E = Duplicates)

R	P	S	N
R1	11% (12)	67% (75)	22% (25)
R2	12% (2)	56% (9)	31% (5)
R3	12% (1)	50% (4)	38% (3)

Q4.1 (E = Duplicates)

R	Detection	P	S	N
R1	ZeroER	5% (3)	61% (34)	34% (19)
	Key Collision	16% (9)	73% (41)	11% (6)
R2	ZeroER	12% (1)	50% (4)	38% (3)
	Key Collision	12% (1)	62% (5)	25% (2)

Q5 (E = Duplicates)

R	Dataset	P	S	N
R1	Airbnb	4% (1)	86% (24)	11% (3)
	Citation	11% (3)	71% (20)	18% (5)
	Movie	29% (8)	21% (6)	50% (14)
	Restaurant	0% (0)	89% (25)	11% (3)

Key takeaways

1. The impact on ML varies across different error types.
2. Cleaning methods and their impact may depend on the datasets used.
3. Model and cleaning algorithm selection play a role in mitigating negative impacts.

Overall Observations

For Single Error Types

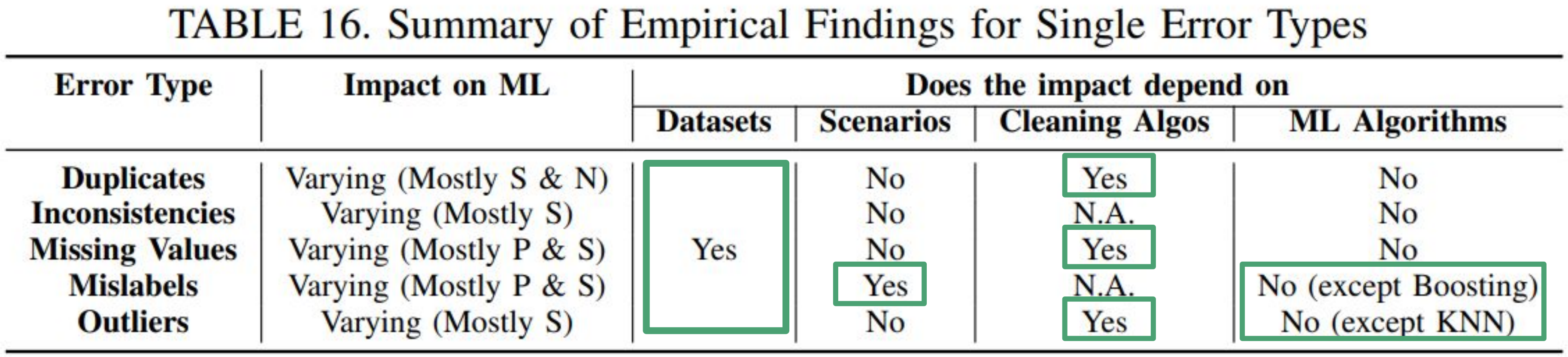


TABLE 16. Summary of Empirical Findings for Single Error Types

Error Type	Impact on ML	Does the impact depend on			
		Datasets	Scenarios	Cleaning Algos	ML Algorithms
Duplicates	Varying (Mostly S & N)	Yes	No	Yes	No
Inconsistencies	Varying (Mostly S)		No	N.A.	No
Missing Values	Varying (Mostly P & S)		No	Yes	No
Mislabels	Varying (Mostly P & S)		Yes	N.A.	No (except Boosting)
Outliers	Varying (Mostly S)		No	Yes	No (except KNN)

Observations

Strong Dependency on Dataset:

- The impact of cleaning on ML is highly dependent on the dataset. Noticeable differences exist between results of general analyses (Q1) and dataset-specific analyses (Q5) for all error types.
- Practitioners should avoid making arbitrary cleaning decisions, as the impact varies across datasets even for the same error type.

Consistent Impact w.r.t. Model and Scenario:

- While some ML models show sensitivity to specific error types (e.g., KNN to outliers, Adaboost to mislabels), there's usually no notable difference in the impact across different ML models.
- Consistent impacts are observed across models, suggesting that if cleaning has a particular impact for one ML model, it is likely to have a similar impact for other models.
- The difference between results of model-agnostic analyses (Q1) and model-specific analyses (Q3) is negligible, emphasizing the broad applicability of data cleaning in ML classification tasks.

Observations

Strong Dependency on Cleaning Algorithms:

- Different datasets may require different cleaning algorithms due to variations in error distributions.
- Despite dataset-dependent differences, consistent patterns are observed when comparing results across different cleaning algorithms (R1, R2, R3).
- Data cleaning is more likely to positively impact better ML models, especially those chosen using a validation set.
- The cleaning algorithm selected using a validation set is more likely to have a positive impact than a randomly chosen cleaning algorithm.
- Selecting a data cleaning algorithm using a validation set is a sensible practice, but it may not guarantee optimal results in all cases.

Cautionary Note:

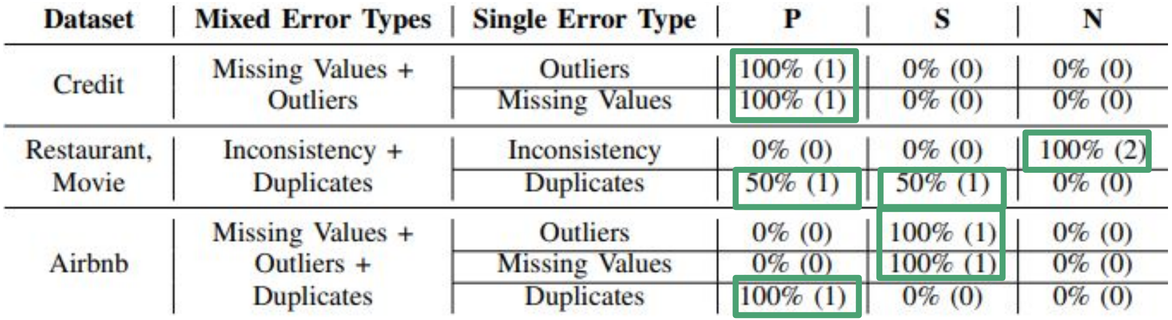
- While validation set-based selection is generally effective, instances exist where the selected cleaning algorithm may still degrade ML model performances.
- Practitioners should approach data cleaning with careful consideration, recognizing that no single automatic cleaning algorithm is universally optimal.

Mixed Errors, RobustML, and Human Cleaning

Cleaning Mixed Error Types

Setup:

- Study the impact of cleaning multiple error types simultaneously.
- Consider a Cartesian product of cleaning algorithms for each component error type.
- Compare the best model obtained by cleaning all error types with that obtained by cleaning a single error type.



Dataset	Mixed Error Types	Single Error Type	P	S	N
Credit	Missing Values + Outliers	Outliers	100% (1)	0% (0)	0% (0)
		Missing Values	100% (1)	0% (0)	0% (0)
Restaurant, Movie	Inconsistency + Duplicates	Inconsistency	0% (0)	0% (0)	100% (2)
		Duplicates	50% (1)	50% (1)	0% (0)
Airbnb	Missing Values + Outliers + Duplicates	Outliers	0% (0)	100% (1)	0% (0)
		Missing Values	0% (0)	100% (1)	0% (0)
		Duplicates	100% (1)	0% (0)	0% (0)

Not always better than cleaning a single error type, similar to the impact observed for cleaning a single error type.The positive impact is dataset-dependent.

Rare instances of negative impact when cleaning inconsistency + duplicates, which can be worse than cleaning inconsistency only.

Cleaning missing values or outliers on top of any single error type is likely to bring positive impacts and has no negative impacts in all cases.

CleanML vs. Robust ML Approaches

Setup:

- Compare CleanML with robust ML approaches.
- Evaluate CleanML against NaCL (robust Logistic Regression for missing values) and deep learning (MLP) models for other error types.
- Use 20 train/test splits for each dataset.

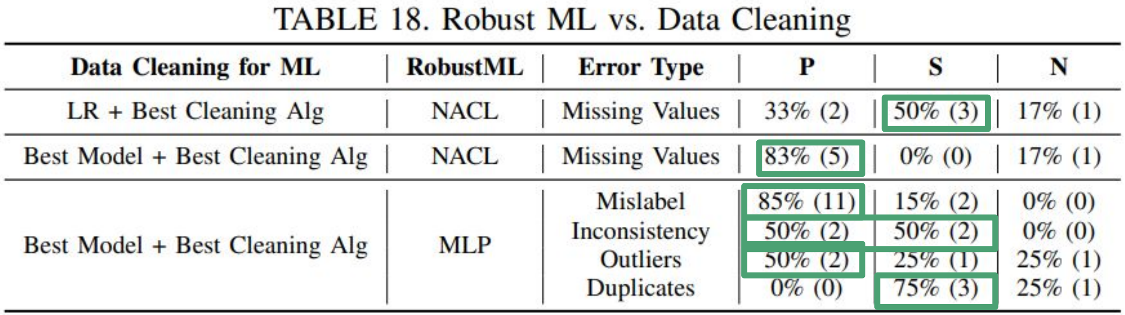


TABLE 18. Robust ML vs. Data Cleaning

Data Cleaning for ML	RobustML	Error Type	P	S	N
LR + Best Cleaning Alg	NACL	Missing Values	33% (2)	50% (3)	17% (1)
Best Model + Best Cleaning Alg	NACL	Missing Values	83% (5)	0% (0)	17% (1)
Best Model + Best Cleaning Alg	MLP	Mislabel	85% (11)	15% (2)	0% (0)
		Inconsistency	50% (2)	50% (2)	0% (0)
		Outliers	50% (2)	25% (1)	25% (1)
		Duplicates	0% (0)	75% (3)	25% (1)

Data cleaning often leads to a better end model compared with robust ML approaches.

The benefit of data cleaning over robust ML widens, highlighting the flexibility of data cleaning for any error type and model.

Duplicates are an exception where deep learning is overall better than cleaning, consistent with the finding that cleaning duplicates is likely to harm ML models.

Human Cleaning vs. Automatic Cleaning Algorithms

Setup:

- Compare human cleaning with automatic cleaning algorithms.
- Evaluate datasets where humans manually filled missing values or corrected mislabels.
- For datasets with inconsistencies, manual curation of data quality rules is compared with automatic cleaning methods.

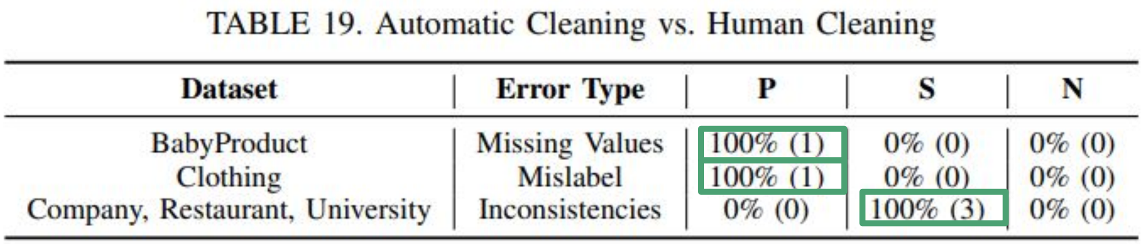


TABLE 19. Automatic Cleaning vs. Human Cleaning

Dataset	Error Type	P	S	N
BabyProduct	Missing Values	100% (1)	0% (0)	0% (0)
Clothing	Mislabel	100% (1)	0% (0)	0% (0)
Company, Restaurant, University	Inconsistencies	0% (0)	100% (3)	0% (0)

Human cleaning is better than the best automatic cleaning method for datasets with missing values and mislabels (BabyProduct, Clothing).

Rule-based cleaning for datasets with inconsistencies shows no significant difference from using automatic cleaning methods.

Future Research

- Explore diverse ML tasks beyond classification.
- Enhance dataset variability for human cleaning studies.
- Develop a theoretical framework for consistent machine learning.
- Optimize automatic cleaning algorithms for ML.
- Address challenges in handling multiple error types.
- Investigate human-centric cleaning solutions.

Quiz Question

What is the key challenge in designing new automatic cleaning algorithms for machine learning?

1. Model uncertainty
2. Lack of ground truth
3. Cleaning algorithm complexity
4. Randomness dilemma

2

Lack of ground truth

THANK YOU