Towards a Framework for the Preparation of High Quality Data for Use by Machine Learning Algorithms

Rasidatou Nabi[1(✉)], Yaya Traoré[1], and Julie Thiombiano[2]

[1] University of Joseph KI-ZERBO, Ouagadougou, Burkina Faso
rasidatou.nabi@ujkz.bf
[2] University of Nazi BONI, Bobo-Dioulasso, Burkina Faso

Abstract. Nowadays, companies and organizations have access to various data collection tools that enable them to amass vast amounts of data, which can be stored in databases. This data can be leveraged by machine learning algorithms to extract valuable information for decision-makers. However, this raw data is often of poor quality, containing errors such as missing data and outliers, requiring the intervention of technicians and domain specialists to prepare the data to ensure the $F1\_Score$ of the analysis. This article proposes a framework for preparing high-quality data for machine learning algorithms, as manually identifying reliable data from a large pool can be challenging and time-consuming. Our approach is an architectural method that combines data preparation techniques to generate dataset quality.

Keywords: Data processing · Quality data · Missing data · Encoding · Normalization

1 Introduction

Presently, many data collection tools have been developed, which allow companies and organizations to collect large amounts of data and store them in databases. Machine learning can analyze data to provide actionable insights for decision-makers. However, the raw data collected is often of poor quality because it can contain inaccurate data, missing data, outliers, and so on. Therefore, a data processing phase is necessary.

Data processing refers to any activity aimed at improving the quality, usability, accessibility, or portability of data. The ultimate goal of data preparation is to enable people and analytical systems to have clean, usable data converted into useful information [1]. Data processing involves analyzing raw data to produce quality results. This includes cleaning missing data and outliers, encoding categorical data, normalizing data, and reducing data through attribute selection.

Several techniques are used to process the data depending on the anomalies that may exist and the types of data. For this purpose, many contributions have been made, but each contribution does not take into account the treatment of all

A. Sere et al. (Eds.): AFRICOMM 2023, LNICST 588, pp. 154–162, 2025.
https://doi.org/10.1007/978-3-031-81573-7_12

anomalies [19]. Therefore, it is imperative to set up a data preprocessing method that integrates the essence of data preprocessing techniques, allowing all those who wish to obtain quality data from the raw data.

In this paper, we introduce an architectural approach that combines various data preprocessing techniques to generate high-quality data for machine learning algorithms. Data quality refers to the condition of data based on factors such as accuracy, completeness, consistency, and reliability. Measuring the level of data quality can help identify potential errors that need to be corrected. The data pre-processing process intervenes in the correction of these errors.

Thus, this paper is structured as follows. Section 2 discusses related work on preprocessing techniques. Section 3 presents our proposed preprocessing approach. Section 4 presents the experimentation of our approach we will end with a conclusion and perspectives in Sect. 5.

2 Related Work

There are many scientific studies on data preparation techniques based on machine-learning algorithms. For each technique, a lot of work has been done to improve data preprocessing. *Potdar and al.* [2] have compared seven different techniques that can be used for encoding categorical variables. The main objective of this study was to classify a dataset using neural networks. To achieve this, the authors made use of a second-hand car dataset. Based on the prediction results, the sum Coding and Backward Difference Coding techniques have provided an accuracy of 95%. Therefore, it can be inferred that these two techniques are most suitable for making predictions that involve a categorical dataset.

According to [3], when analyzing data, we often face a loss of information due to missing values. To tackle this problem, several techniques have been developed, such as imputation techniques, which help in substituting the missing data. The authors have presented several methods for imputing the missing data, and analysis of the processed data using these techniques has proven to improve the accuracy of the model. In the analysis of data, we find losses of information due to the presence of missing values. To remove these missing data, various techniques have been explored, such as the use of imputation techniques, and the authors have presented various methods for imputing missing data. Analysis of the data processed using imputation techniques helps to improve model accuracy [3] and [4,13,14,16].

Pandey and al. [5] describe two normalization techniques that they implemented. The authors use these techniques to create a global classification of IRIS data and measure the accuracy of their approach using the cross-validation method and the R programming language. Specifically, the article focuses on two normalization approaches: Z-Score normalization and Min-Max normalization. *Cousineau and al.* [6] discuss different methods for detecting possible outliers. These techniques can be divided into two categories: those that work with univariate data and those that work with multivariate data. The work will evaluate each case and provide appropriate recommendations.

According to the research conducted by *Motoda and al.* in [7], feature selection is a crucial process that aims to eliminate some of the initial features and keep only the most relevant ones. This can be achieved by using a specific criterion to optimally reduce the feature space. On the other hand, feature extraction refers to the process of generating new features that can be utilized independently or in conjunction with others.

Inspired by these methods, we propose an architectural method that gathers data preprocessing methods to produce quality data. In the next section, our method which is the combination of the different methods contained in the above-mentioned works is described.

3 Approach to Data Preprocessing

The Fig. 1, below presents our method of data preprocessing which is based on the different techniques that have been defined in the existing work.

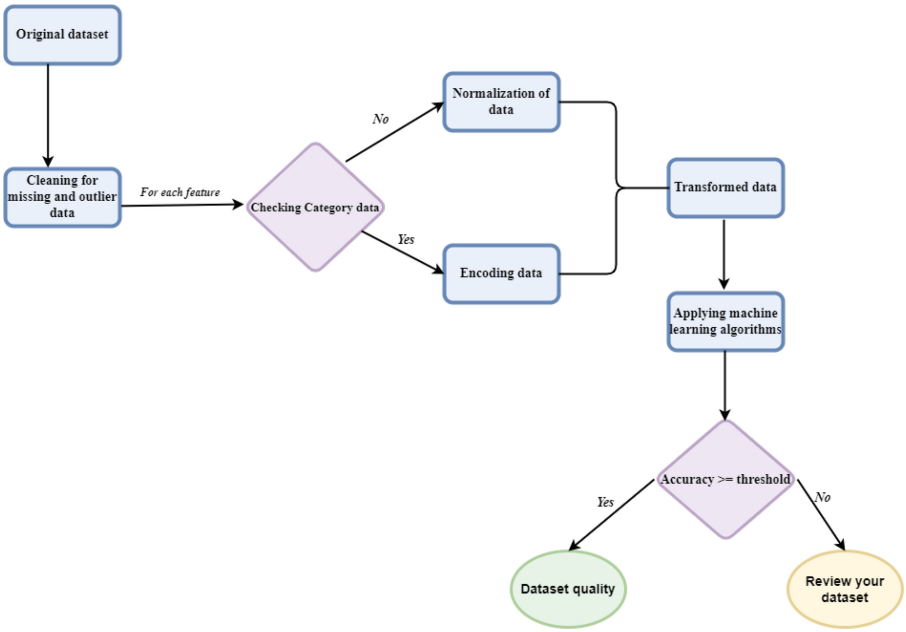

Fig. 1. approach of data preprocessing

– **step 1:** Handling missing and outlier data is a crucial step in preparing data for analysis and modeling. In our approach, two techniques are proposed for handling missing and outlier data. For the treatment of missing data, the mean imputation technique is proposed. This technique involves replacing

missing values with the mean of non-missing values of the same variable. This technique is widely used in the scientific community for its simplicity and robustness. For the treatment of outliers, the interquartile range technique is proposed. This technique detects outliers by calculating the difference between the first and third quartiles of a variable. Values outside this range are considered outliers and replaced by the variable's median value [8,12]. It should be noted that these techniques are not the only ones available for handling missing and outlier data.

- **step 2:** Data normalization and encoding are also two important steps in data preparation. In our approach, two techniques are proposed for normalizing and encoding data. The process begins by checking whether categorical data are present for each feature in the dataset. When the selected feature contains categorical data, the hot coding technique is proposed to transform the data. This technique involves creating binary variables for each category of the categorical variable. This technique is widely used in the scientific community to encode categorical data due to its simplicity and its ability to avoid unnecessary weighting of categories. When the selected feature does not contain categorical data, normalization is performed. For normalization, the min-max normalization technique is proposed. This technique involves transforming the data to fall within a specific range of values, usually between 0 and 1. This technique is widely used in the scientific community to normalize data due to its simplicity and robustness [9]. After normalization and encoding, the data is transformed and ready to be used for analysis and modeling. It should be noted that these techniques are not the only ones available for data normalization and encoding.

$$X_{normalized} = \frac{\chi - min(\chi)}{max(\chi) - min(\chi)} \tag{1}$$

where: min and max are the minimum and the maximum value of χ

- **step 3:** Data quality testing is also an essential step in data preparation. In this paper, an approach is proposed to check that data have been properly cleaned. For this, models based on machine learning algorithms such as support vector machines (SVM), Naive Bayes, K-Nearest-Neighbors Classifier XG Boost Classifier, and random forests are used. This evaluation is carried out using the input data transformed to form these models. Model performance is then measured against parameters such as the $F1\_Score$ [10,17]. In this approach, if the $F1\_score$ is greater than or equal to a defined threshold, the data are considered to be of high quality. However, if the $F1\_score$ is below the defined threshold, the data quality testing process starts again.

The following Algorithm 1 is used to describe our methodology.

Algorithm 1. Algorithm used by the High-Quality Data Preparation framework

1: **Input**: D: Original dataset with C_n columns
2: ML:Machine learning algorithm
3: nval: number of folds of the cross-validation
4: σ: Threshold
5: **Output** : QD : Quality data
6: **Begin**
7: $D \leftarrow MissingData(D)$
8: $D \leftarrow Outliers(D)$
9: **for all** column C in D **do**
10: **if** C is numeric data **then**
11: $C \leftarrow Normalizer(C)$
12: **else**
13: $C \leftarrow Encoder(C)$
14: **end if**
15: **end for**
16: $F1\_score \leftarrow TrainML(ML, D, nval)$
17: **if** $F1\_score \geq \sigma$ **then**
18: $QD \leftarrow D$
19: **else**
20: Review your dataset
21: **end if**
22: **End**

The Algorithm 1 illustrates the algorithm used by the High-Quality Data Preparation framework for preparing high-quality data. Lines 1 to 4 specify the input data, which includes the dataset, quality threshold, and Machine Learning algorithms. Line 5 indicates that the expected output is high-quality data. In line 7, we first address missing data, and in line 8, we utilize the previously processed dataset with missing data to handle outliers [20]. From line 9 to line 15, we iterate through each column of the dataset, normalizing numerical data (line 11) and encoding categorical data (line 13). We now have a prepared dataset ready to be used in the learning process. In line 15, the function train is used to train the model [21]. We use the Cross-validation to evaluate the model. The cross-validation model randomly divides the training data into *nval* folds ($nval = 10$). In each iteration of the dataset, the cross-validation model uses one fold as the validation dataset. It uses the remaining $nval - 1$ folds to train a model. Each of the *nval* models is tested against the data from all the other samples. In line 16, the function TrainML is used to train the model. We consider 70% of the prepared dataset as the training data, leaving 30% for testing purposes. The $F1\_score$ (line 16) metric is used to assess the performance of the model and to decide whether the data are of high quality or not. the function TrainML repeats the training of the model *nval* times. In lines 16 to 20, if $F1\_score$ is greater than or equal to the threshold specified we return high-quality data else we prompt the user to review their dataset.

4 Experimentation of Approach

4.1 Programming Language and Framework

To facilitate and accelerate the implementation of our solution, we chose the Python programming language and the framework Streamlit. Python is a programming language that respects object-oriented programming paradigms. Consequently, as a Python-based development framework, streamlit was chosen. Streamlit is an open-source platform that enables the creation of data apps with Python, using Python scripting APIs, widgets, and instant deployment. The platform integrates various libraries such as Scikit Learn, OpenCV, vega-Lite, PyTorch, Numpy, Seaborn, Deck GL, Tensorflow, Python, Matplotlib, and Pandas, which can be helpful for machine learning engineers to create Python-based applications.

4.2 Execution Environment

Our work setup consists of a laptop running Ubuntu Operating System with the following specifications.

- Processor: Intel(R) Core(TM) i5-8259U CPU @ 2.30GHz, 4 cores and 8 logic processors
- RAM: 8 Gb
- Graphics: Intel(R) Iris(R) Plus Graphics 655

4.3 Presentation of the Dataset

We have decided to use the Iris flower dataset to test our data processing approach. This dataset is a multivariate set that was made famous by the British statistician and biologist. It comprises 50 samples from each of the three Iris species (Iris setosa, Iris virginica, and Iris versicolor). Four features were measured from each sample, including the length and width of the sepals and petals in centimeters. You can access the Iris dataset through the provided link https://www.kaggle.com/datasets/saurabh00007/iriscsv/download?datasetVersionNumber=1.

4.4 Results and Discussion

This section presents the results of fitting our models to the dataset. We measured the performance using the $F1\_score$. The $F1\_score$ considers precision and recall, two important metrics for classification models. Precision measures the number of true positives divided by the total number of positive predictions, while recall measures the number of true positives divided by the total number of true positives and false negatives. The $F1\_score$ is a more reliable measure than accuracy because it takes into account both precision and recall [11].

Also, for quality checks, we set the threshold value at 80%. It should be noted that other performance measures can also be used depending on the specifics of the data and the situation.

Table 1, shows the results ($f1\_score2$) obtained after testing our dataset with different algorithms.

Table 1. Model Evaluation Results.

algorithms	$f1\_score0$ (%)	$f1\_score1$ (%)	$f1\_score2$(%)
Support Vector Machine (SVM)	61.33%	90.85%	93.55%
Random Forest Classifier	73.66%	91.53%	95.58%
XG Boost	75.55%	91.35%	95.58%
Naive Bayes	76.65%	91.53%	95.58%
K-Nearest-Neighbors Classifier	72.65%	92.43%	94.03%

Let us consider:

- $f1\_score0$ Results before applying our processing method,
- $f1\_score1$ Results after applying our processing method without normalization and encoding,
- $f1\_score2$ Results after applying our processing method with normalization,

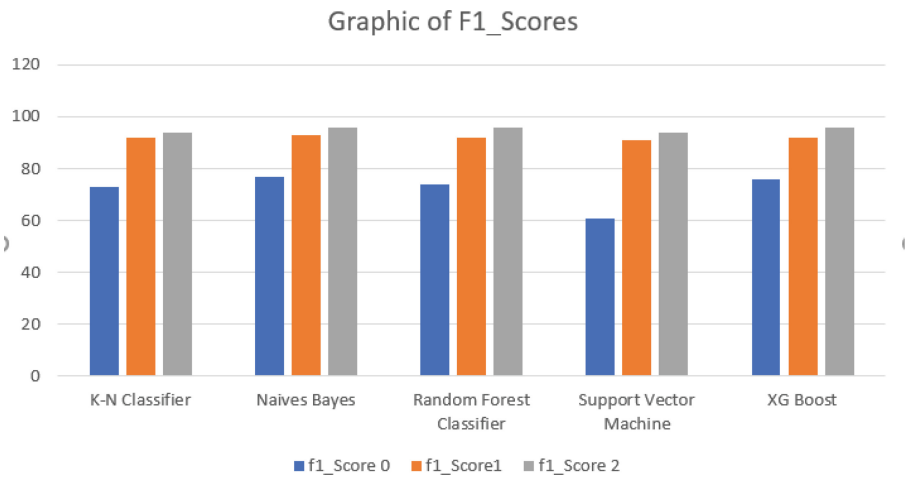

Fig. 2. Graphic of $F1\_scores$ (%)

Table 1 shows firstly the results obtained without applying our data processing method ($f1\_score0$), secondly the results obtained by applying our data

processing method without normalization or encoding ($f1\_score1$), and finally by applying our complete data processing method ($f1score2$). The complete data preprocessing technique has been utilized to enhance the classification models performance, which is measured using the $F1\_score$. Techniques like normalization and coding have been employed to enhance the quality of the input data [15]. A comparative analysis of the model's performance before and after the data preprocessing can assist in determining the effectiveness of this method. However, this technique may not be suitable for all types of data. Nonetheless, by applying data preprocessing techniques, the performance of classification models can be significantly improved (Fig. 2).

5 Conclusion

In this paper, we propose a framework that addresses quality data issues and aims to generate high-quality datasets for machine learning algorithms. The framework employs an architectural approach that combines different data preparation techniques. These techniques include handling missing data, encoding, and normalization. The results of our implementation show satisfactory results.

While the results obtained from the Iris Dataset are encouraging, it would be valuable to assess the performance and the robustness of the proposed data pre-processing framework on a wider range of datasets, under different scenarios, including noisy data, imbalanced datasets, and varying levels of data quality.

References

1. Garcia, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining, pp. 59–105. Springer (2015)
2. Potdar, K., Pardawala, T.S., Pai, C.D.: A comparative study of categorical variable encoding techniques for neural network classifiers. Int. J. Comput. Appl. **175** (2017)
3. Luengo, J., García, S., Herrera, F.: On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowl. Inf. Syst. **32**, 77–108 (2012)
4. Song, Q., Shepperd, M.: A new imputation method for small software project data sets. J. Syst. Softw. **80**, 51–62 (2007)
5. Pandey, A., Jain, A.: Comparative analysis of KNN algorithm using various normalization techniques. Int. J. Comput. Netw. Inf. Secur. **9**, 36 (2017)
6. Cousineau, D., Chartier, S.: Outliers detection and treatment: a review. Int. J. Psychol. Res. **3**, 58–67 (2010)
7. Motoda, H., Liu, H.: Feature selection, extraction, and construction. Commun. IICM (Institute of Information and Computing Machinery, Taiwan) **5**, 67–72 (2002)
8. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**(2), 85–126 (2004)
9. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)

10. Kelleher, J.D., Tierney, B., Tierney, B.: Data Science An Introduction. CRC Press (2018)
11. Powers, D.M.: Evaluation: from precision, recall, and F-measure to ROC, informedness, markedness, and correlation. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)
12. Schafer, J.L., Graham, J.W.: Missing data: our view of the state of the art. Psychol. Methods **7**, 147 (2002)
13. Ford, B.L., et al.: Missing data procedures: a comparative study. Sampling Studies Section, Sample Surveys Research Branch, Statistica (1976)
14. Wayman, J.C.: Multiple imputation for missing data: what is it and how can I use it. Annual Meeting of the American Educational Research Association, Chicago, IL, vol. 2, p. 16 (2003)
15. Ridzuan, F., Zainon, W.M.N.W.: Diagnostic analysis for outlier detection in big data analytics. Procedia Comput. Sci. **197**, 685–692 (2022)
16. Gleason, T.C., Staelin, R.: A proposal for handling missing data. Psychometrika **40**, 229–252 (1975)
17. Acuna, E., Rodriguez, C.: A meta analysis study of outlier detection methods in classification. Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez, vol. 1, p. 25 (2004)
18. Liu, H., Motoda, H.: Data reduction via instance selection. In: Instance Selection and Construction for Data Mining, pp. 3–20. Springer (2001)
19. Chen, J., Shao, J.: Nearest neighbor imputation for survey data. J. Official Stat. **16**, 113 (2000)
20. Sinsomboonthong, S.: Efficiency comparison in prediction of normalization with data mining classification. Diabetes **768**, 231 (2021)
21. Livingston, F.: Implementation of Breiman's random forest machine learning algorithm, ECE591Q Machine Learning Journal Paper, pp. 1–13 (2005)