AUTOMATED DATASET PREPROCESSING

Surbhi Bajirao[1], Shivam Mahajan[2], Rahul Sangamnere[3], Robin Singh[4], S.V. Patil[5]

[1] *Student, Department of Computer Engineering, Sinhgad college of engineering, Maharashtra, India*
[2] *Student, Department of Computer Engineering, Sinhgad college of engineering, Maharashtra, India*
[3] *Student, Department of Computer Engineering, Sinhgad college of engineering, Maharashtra, India*
[4] *Student, Department of Computer Engineering, Sinhgad college of engineering, Maharashtra, India*
[5] *Professor, Department of Computer Engineering, Sinhgad college of engineering, Maharashtra, India*

ABSTRACT

Machine Learning (ML) has grown rapidly in recent years, with many resources available for people to become ML practitioners. Companies are profiting from ML analysis and predictions, and ML Engineers are in high demand and well-compensated. ML has become more prevalent and easier to understand. One crucial stage in ML is Data preprocessing and feature extraction. Data preprocessing involves various tasks to accurately prepare the provided data. From handling missing values to encoding and normalization, each step is important and requires expertise. The preprocessing steps depend on the type of data, such as categorical data, continuous data, or even image data. Learning and becoming an expert in all the cleaning steps can be challenging and time-consuming, with no guarantee of desired results. This is where automation can help. Our goal is to automate the entire data preprocessing process, making it easier and more productive for ML Engineers. Users will only need to provide the dataset without manually selecting processing techniques, as done in the latest Data Mining tools. Our application will analyze the dataset and automatically apply suitable techniques. With automation, even people unfamiliar with ML concepts can preprocess the dataset. This opens up opportunities for individuals from various domains who want to perform ML operations but lack expertise in preprocessing. By automating data preprocessing, we aim to simplify the work for ML Engineers, increase productivity, and make ML more accessible to a wider range of users.

Keyword: - *Auto-ML, Encoding, Algorithm, Big Data, KNN.*

1. INTRODUCTION

In recent years, machine learning models have been used in various areas to solve complex tasks. However, developing these models manually requires time and resources, increasing the demand for machine learning experts. To reduce these development costs, automated machine learning (Auto-ML) has emerged as a concept. Auto-ML aims to decrease the demand for data scientists and enable domain specialists to automatically develop machine learning applications without prior ML knowledge.

Auto-ML consists of several core processes, including data preprocessing, feature engineering, model generation, and model evaluation. Among these processes, data preprocessing is crucial and can take up to 50-80 percent of the analysis time. Properly preprocessing the dataset is essential because it can significantly impact the results. Even with the best learning model, if the dataset is not correctly prepared, the model may perform poorly and achieve low accuracy. To make data preprocessing easier for unskilled users, an efficient method is needed. Currently, existing automated techniques for data preprocessing are not mature enough and require significant human intervention. These techniques allow users to implement different preprocessing algorithms but do not consider which one is most suitable for the given dataset to improve accuracy. In our research, we focus on automating all the data preprocessing tasks mentioned above. The goal is to develop an approach that simplifies these tasks and improves model performance. The functionalities of our approach include:

1. Automatic detection of duplicate rows
2. Automatic detection of Features data types
3. Automatic Missing Data Imputation
4. Automatic Categorical Features Encoding
5. Automatic Feature Reduction
6. Automatic Feature Scaling

1.2 LITERATURE SURVEY

Data Preprocessing: Every day huge amount of data is created. Machine learning (ML) can learn and predict from these datasets to make them more valuable. However, a significant issue is that real life data is hardly ever clean, and poor quality of data can have a significant impact on the performance of learning algorithms. As necessary and unavoidable as data pre-processing is, it is a monotonous and inconvenient process. Data scientists normally spend over half of their analytical time on pre-processing, nonetheless non-experts. Therefore, data scientists are keen to develop a technique to automate this procedure. Data pre-processing includes a variety of tasks such as cleaning, encoding, scaling and dimensionality reduction. The task of resolving data issues is referred to as data cleaning. Typical data issues include missing values, inaccurate datatypes, and repeated rows. Data cleaning is intended to clean the data with missing values, inconsistencies, and noisy data. Data pre-processing is also intended to perform feature encoding, scaling, and Dimensionality reduction.

1) Mehwish Bilal, Ghulam Ali, Muhammad Wasim Iqbal, Muhammad Anwar "Auto-Prep: Efficient and Automated Data Pre-processing Pipeline", 16 Aug 2022.

Detection of Data Types - Detection of data types beforehand, can help to analyze and evaluate preprocessing and encoding options for each respective feature. Despite the diversity of data types, all the datatypes are not equally significant in the field of ML. The statistical data types within a data contain a wealth of information that is particularly valuable in the context of ML. Several feasible techniques are present to detect data types from a data. Some methods are straightforward and might only need few heuristics or statistics. But some complex and advanced techniques are also available which employ machine learning models for datatypes detection. Messy tables is a python package, uses brute force to predict data types. Brute force guessing extract a random sample from an attribute and then attempt to convert each value in the sample to every possible data type. The successful transformations rate is calculated for each data type, and the most likely data type for that column is determined by a majority vote. Messy tables consider the following data types: Integer, String, Decimal, Data, and Bool. This method is adaptable and simple to implement.

2) Mehwish Bilal, Ghulam Ali, Muhammad Wasim Iqbal, Muhammad Anwar "Auto-Prep: Efficient and Automated Data Pre-processing Pipeline", 16 Aug 2022.

Missing Data Imputation - In practice, missing data is a frequent occurrence because of manual data entry systems, incorrect measurements, equipment malfunctions, intentional omissions et cetera. A few missing instances in some features can significantly reduce the sample size. Subsequently, the efficiency and precision of data analysis can be compromised, weaken the statistical power, and the parameter estimation could be biased because of the differences between complete and missing data. In the field of ML, the missing values can increase the classifier error rate. Therefore, missing data must be addressed before employing learning models. Whether data comes from experiments, surveys, or secondary sources, missing data is abundant. But what effect does this have on statistical analysis results? That is contingent upon two factors: the mechanism that caused the data to be missing and how the data analyst handles it. In any type of study, data may be missing because of an accident or a data entry error. A meticulous understanding of data enables us to ascertain the mechanism of missing data. Missing data Mechanisms can be divided into three categories which are, MCAR, MAR and MNAR. Collectively there are numerous viable techniques to deal missing data. Different methods are appropriate for different conditions. Deletion is effective only in MCAR without producing a significant bias. Statistical information-based imputation is essentially effective to MCAR only because they estimate data without contemplating the relationship between attributes, MICE, matrix factorization, and KNN are mostly effective in MAR. Some other alternative missing imputation strategies are also available, including missing

indicator and maximum likelihood. Those approaches are not explained here because either they are too complex to automate or can only be useful to MCAR.

3) Mehwish Bilal, Ghulam Ali, Muhammad Wasim Iqbal, Muhammad Anwar "Auto-Prep: Efficient and Automated Data Pre-processing Pipeline", 16 Aug 2022.

Qualitative Data Encoding - ML algorithms expect all inputs and output attributes to be numerical. This means if a dataset has categorical data, first encode that into numerical format before employing a ML algorithm. Encoding is a mandatory pre-processing stage when working with qualitative data for ML models and there exists a spectrum of methods for categorical data encoding. The appropriate method can have a substantial effect on the performance of a model. One way to identify a determined technique is it will generate the same encoded values every time we employ it, contrary to algorithmic methods. Additionally, these methods have a minimal complexity in terms of run time. Label encoding is essentially the process of allocating a numeral value to each potential value of a categorical attribute. Duan compares the ability of several classifiers discover that to encode qualitative features, one-hot encoding provides satisfactory outcomes. One-hot encoding approach needs very little effort to implement but drawback of this technique is, if we store encoded values directly, it uses a lot of storage resources. For large cardinality, the feature space can soon explode, and we are forced to combat with the curse of dimensionality, but the advantage of One-hot encoding is that it is easy to employ and has an effective running time.

2. PROBLEM STATEMENT

A simple Python-based Auto pre-processing architecture for Automated Machine Learning will offer automated, interactive, and data-driven support to help the users perform data pre- processing tasks efficiently. The suggested method provides valuable insights into a dataset and can handle standard data pre-processing tasks adeptly. Initially, it detects the data problem and presents it to the end-user using compelling visualizations. Then, it optimizes with the most effective data cleaning and preparation method to the user after evaluating the state-of-the-art candidate techniques.

3. METHODOLOGY

In our research, we have researched a Python-based method to make data preprocessing easier and more efficient for users. Our method is designed to automatically analyze the dataset's features and apply the most suitable techniques to improve data quality for better machine learning model performance. We have incorporated existing approaches to address various data cleaning and feature engineering challenges. Our main focus is on automatically detecting the data type, filling in missing values, and encoding and scaling features. We also consider dimensionality reduction to handle high-dimensional data. Our method is designed to handle each task independently and make rational decisions based on the specific dataset. We discuss each subtask in detail, explaining how we tackle each problem effectively. Overall, our goal is to provide an automated, interactive, and data-driven solution that streamlines the data preprocessing process, leading to improved results when using machine learning models.

3.1. AUTOMATIC DATA TYPES DETECTION

To effectively analyze and process data, it is important to have a good understanding of how data is stored and manipulated. In the field of machine learning, the statistical datatype of features plays a significant role. Our goal is not only to identify basic datatypes but also to differentiate statistical datatypes. To achieve this, we propose a strategy that combines a straightforward logical approach with the Pandas library. Pandas is a powerful tool that can automatically detect general datatypes for each feature. However, the datatype inference of Pandas is limited to int, float, and object, which can be a drawback. In practice, we often encounter situations where Pandas classifies a large number of features as 'objects'. For example, it may not automatically recognize datetime columns unless explicitly specified. This can lead to inconsistencies in data handling. By leveraging our proposed strategy, we aim to overcome these limitations and improve the accuracy of datatype detection, including the proper recognition of datetime columns. This will ensure more reliable data processing and analysis in machine learning tasks.

3.2. AUTOMATIC MISSING VALUE IMPUTATION

To handle missing values in our data, we follow a step-by-step approach. Firstly, we need to identify the missing values in our dataset. It's important to note that no single algorithm is always the best for this task, as its effectiveness depends on the type of missing values we have. To tackle this challenge, we break it down into several subtasks. First, we discover where the missing values are located in the dataset. Next, we visualize the missing data using graphs and charts to help users understand the extent and patterns of missingness. This visual representation is important for gaining insights into the missing ratio and the underlying missing mechanism. After visualizing the missing data, we proceed to address the missing values. We offer multiple techniques for imputing missing values, and we evaluate their performance. The goal is to recommend the most suitable technique based on the specific dataset and missing pattern. To provide a clear overview of the workflow, we present it in Figure 2, which gives a high-level representation of the steps involved. By following this structured approach, we can effectively handle missing values and make informed decisions on the best imputation technique to use.

3.3. DISCOVER MISSING VALUES

The first challenge to handle the missing data automatically is to detect the presence of missing Values. Missing values may be represented in the dataset using a variety of alternative formats, including '?' and 'nan'. We have categorized the formats into three types for better understanding. (I) The representations which can be treated by default as null values, such as: 'NAN', 'n/a', and Empty cells are Standard Missing Values. (II) Some datasets tend to have non-standard representation of missing values as well, such as: '—', '-', 'na', or '?', etc. (III) Depending on the context of the dataset, certain datasets represented missing values in an unexpected manner. For example, if a feature's range is 0 to 100, then 9999 may indicate the missing value. We've constructed our unique missing values list as default for the first two scenarios, but for the third scenario, let's say, we cannot take 9999 as missing/null for every dataset. As a result, it would be preferable to identify missing values in an interactive manner. To be more particular, in addition to the most common missing characters, such as 'nan,' we ask the user every time before the detection whether they want to add any additional specific value to be identified as missing.

3.4. MISSING MECHANISM

We have already established from literature that there exist three kinds of missing mechanisms: MAR, MCAR, and MNAR. Unfortunately, the hypothesis in case of MNAR cannot be tested because the necessary information is unavailable. We may have good cause to believe that the prospect of missingness is influenced by the values that are missing from the data set. For instance, some people may be less prone to record their incomes if they have a high salary, but there's no way to know from the dataset if this is true or not. In case of MAR and MCAR, we can identify if any association exist among the missing values of one variable or feature with some other features. We may not detect any association between the two features, yet a MAR may nonetheless exist because a value might perhaps be absent as a function of several other features.

3.5 IMPUTE MISSING VALUES

The several approaches to clean missing values may be perplexing to a novice user. As a result, this proposed method must use the most efficient strategy to cleaning missing values from each feature. Diverse techniques are appropriate for various missing mechanisms. The most frequently used techniques are addressed in this proposed method: mean, median, mode, most frequent, and k neighbor (KNN), or interpret missing values as a separate category and multiple imputation. These approaches are implemented using scikit learn.

4. CHALLENGES

- Developing a project on automated dataset preprocessing and visualization can be accompanied by several challenges. Understanding and addressing these challenges is crucial for successful project completion. Here are some common challenges may encounter:

  Data Quality and Complexity: Real-world datasets often exhibit various data quality issues, including missing values, outliers, inconsistencies, and noisy data. Devising automated techniques to handle such complexities and ensuring data quality throughout the preprocessing phase can be challenging.

- Algorithm Selection: Choosing the most appropriate preprocessing algorithms and visualization techniques for a given dataset and analysis objective can be difficult. The effectiveness and efficiency of the chosen algorithms need to be evaluated to ensure they meet the project requirements.
- Scalability: As datasets grow in size and complexity, the computational resources required for automated preprocessing and visualization can become a challenge. Developing techniques that can handle large-scale datasets efficiently and effectively is crucial.
- Domain-Specific Challenges: Different domains have unique data characteristics and requirements. Developing automated preprocessing and visualization techniques that are domain-specific and can cater to the specific needs of the application area can be demanding.

6. CONCLUSION

Big data is expanding rapidly across various fields, bringing opportunities for data-driven marketing and advancements in different sectors. However, raw data often contains noise, inconsistencies, and irrelevant information. To extract valuable insights and patterns from this data, it is crucial to preprocess it before analysis. In our work, we automate the data preprocessing tasks to make them more efficient. We explore common challenges in handling data and existing approaches to address them. To support our research, we are developing an automated, data-driven, and interactive system that can identify potential issues in the data and provide results and recommendations to the user. This tool is specifically designed for machine learning applications. It automates important components of data preprocessing, such as detecting data types, handling missing values, encoding qualitative data, scaling features, and selecting and extracting relevant features. We evaluate the performance of models trained on automatically preprocessed data compared to manually preprocessed data by measuring their accuracy. We observe a significant improvement in accuracy when using our automated preprocessing approach. By automating the preprocessing tasks, our approach not only saves users from the tedious work of data preparation but also enhances the accuracy of the resulting models. Furthermore, our Auto-Prep method has the potential for future expansion, allowing for more advanced functionalities and capabilities.

7. REFERENCES

[1]. R. Budjač, M. Nikmon, P. Schreiber, B. Zahradníková, and D. Janácová, ‘‘Automated machine learning overview,’’ Vedecké Práce Materiálovotechnologickej Fakulty Slovenskej Technickej Univerzity v Bratislave so Sídlom v Trnave, vol. 27, no. 45, pp. 107–112, 2019.
[2]. H. J. Escalante, ‘‘Automated machine learning—A brief review at the end of the early years,’’ 2020, arXiv:2008.08516.
[3]. A. Truong, A. Walters, J. Goodsitt, K. Hines, C. B. Bruss, and R. Farivar, ‘‘Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools,’’ in Proc. IEEE 31st Int. Conf. Tools Artif. Intell. (ICTAI), Oct. 2019, pp. 1471–1479.
[4]. R. Elshawi, M. Maher, and S. Sakr, ‘‘Automated machine learning: State-of-the-art and open challenges,’’ 2019, arXiv:1906.02287.
[5]. A. P. Barata, F. W. Takes, H. J. van den Herik, and C. J. Veenman, ‘‘Imputation methods outperform missing-indicator for data missing com pletely at random,’’ in Proc.
[6]. G. Chhabra, V. Vashisht, and J. Ranjan, ‘‘A comparison of multiple imputation methods for data with missing values,’’ Indian J. Sci. Technol., vol. 10, no. 19, pp. 1–7, 2017.
[7]. Mehwish Bilal, Ghulam Ali, Muhammad Waseem Iqbal, Muhammad Anwar, Muhammad Sheraz Arshad Malik 4, And Rabiah Abdul Kadir, “Auto-Prep: Efficient and Automated Data Preprocessing Pipeline” Digital Object Identifier 10.1109/ACCESS.2022.3198662