International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307 OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT25112854

The Impact of Data Preprocessing on Machine Learning Model Performance: A Comprehensive Examination

Everleen Nekesa Wanyonyi*[1], Newton Wafula Masinde[2]

*[1]Department of Computer Science, Murang'a University of Technology, Murang'a, Kenya

[2]Department of Computer Science, Jaramogi Oginga, Odinga University of Science and Technology, Bondo, Kenya

ARTICLEINFO

Article History:

Accepted : 25 April 2025

Published: 27 April 2025

Publication Issue

Volume 11, Issue 2

March-April-2025

Page Number

3814-3827

ABSTRACT

Machine Learning (ML) models have been extensively applied in various fields to enhance prediction. For instance, in cybersecurity, they examine large amounts of data, establish trends in the data, and draw insights from previous events, to enhance detection and respond to cyber threats. Random Forest, Logistic Regression, K-Nearest Neighbor and LSTM are some of the popular ML models vastly used for anomaly detection. The accuracy of these models is therefore the cornerstone of the organization's information systems security since wrong predictions result to false positives and negatives which significantly reduce employees' output and may result into workers' frustrations when interacting with the information systems. Among the many factors that affect ML model performance, data pre-processing has been underscored. Using the various publicly available datasets, this paper examines the impact of data preprocessing techniques on selected ML model architectures' performance. Training time, Accuracy, Precision, Recall and F1 scores are used for evaluating the ML models' performance.

Keywords: Data preprocessing, Machine Learning, feature engineering, data imbalance, data cleaning, transformation

Introduction

ML has transformed a number of industries, including healthcare, banking, Natural Language Processing (NLP), and computer vision. This is because automation through Artificial Intelligence (AI) and ML algorithms has several important benefits, including improved productivity, time and cost effectiveness, less human error, accelerated business decision-making, consumer preference forecasts, sales optimization, and handling complex interrelationships between variables (Borodkin et al., 2023). ML models rely on data for training and validation, and therefore, the quality and organization of these data has a significant impact on their performance (Frye et al., 2021). For raw data to be converted into a format that improves model performance, generalization, and efficiency, data preprocessing is a critical stage in the ML pipeline (Amato & Di Lecce, 2024).

Data preprocessing stage takes the unprocessed data and transforms it into a format that can be interpreted and analyzed by computers and ML algorithms. Usually, this process requires a significant amount of time and effort. In most cases, up to 80% of ML development is taken by data preprocessing. This therefore takes more time which could have been used by model training and validation. Real-world data is rarely error-free because many problems that may not have been identified at the time the data was acquired, such as sensor failure, data transmission problems, or improper data input, might result in corrupt data (Brijith, 2023).Due to this setback, the preprocessing step takes more time during development because of the many activities applied to the data to improve its quality. It is therefore important to identify the most influential preprocessing activities to apply to the data so as to reduce time and focus more on the training and validation steps.

In order to evaluate the impact of preprocessing on ML model reliability, training stability, and computing efficiency, this experiment examines numerous significant preprocessing methods, such as data cleaning, data imbalance correction, feature scaling, and dimensionality reduction. We highlight recommended approaches and potential problems in data preprocessing by examining experimental outcomes from various datasets and ML architectures. Optimizing ML learning applications requires an understanding of the importance of data preprocessing. By providing insights into how preprocessing selections can have a substantial impact on model outcomes, this study seeks to close the gap between theoretical understanding and real-world application.

Therefore, the goal of this experiment is to test the effectiveness of various data preprocessing activities on various ML models' performance using different datasets. A ranking of the most effective data preprocessing techniques is done to help ML model developers to easily select the most effective given the limited development time. The rest of the paper is divided as follows. Section 2.0 highlights related works; 3.0 discusses common data preprocessing techniques; 4.0 shows the experimental setup; 5.0 presents results and discussion while 6.0 gives a conclusion.

A. Related Works

The literature on how data preprocessing affect ML model performance is examined in this section. We focus on various data preprocessing techniques and how they affect performance of different kinds of ML algorithms.

A substantial amount of research focus on the distinct impacts of particular preprocessing methods. For instance, the effect of data cleaning techniques (such as handling missing values and detecting and removing outliers) on model accuracy has been studied (Lee et al., 2021). It can be established that proper cleaning can greatly enhance performance, particularly in datasets with high noise levels. Li et al., (2021) extend their research on the impact of data cleaning on ML model performance by conducting an experiment to establish how exactly data cleaning affects ML model performance. According to the study, missing values, outliers, duplicates, inconsistencies, and mislabels are among the five error types that are common in real-world datasets that need to be corrected. The authors established that correction of each of the errors has a different impact on the algorithms' classification tasks. Despite the study's broad focus on data cleaning, other data preprocessing requirements are neglected yet they also have a big impact on data quality.

In addition to data cleaning, extensive research indicate how important data transformation is to ML model performance. For instance, research on data transformation impact on performance of software defect prediction has demonstrated that raw data frequently possesses properties that can significantly reduce the effectiveness of numerous ML models (Zhao et al., 2022). Skewed distributions, outliers, different feature sizes, and non-linear correlations are some examples of these traits (Zheng & Casari, 2018). By addressing these problems, data transformation attempts to improve the data's suitability for modelling and may increase the precision, effectiveness, and interpretability of the model (Liew et al., 2024). Zhao focuses on data transformation for regression models neglecting ML models tasks such as clustering and classification.

The challenges presented by imbalanced datasets-where the number of examples in one or more classes is substantially fewer than the number of instances in the majority class(es) are the subject of a substantial amount of research. As evidenced by studies such as Balla et al., (2023), training ML models on imbalanced data frequently results in biased predictions, with models favoring the majority class and underperforming on the minority class(es), which are frequently the most important. A variety of class imbalances, such as the overlapping class distributions and the small disjoints problem, were examined in along with their effects on ML model learning (Dandu et al., 2024). Additionally, the effects of unbalanced data have been well documented in a variety of real-world applications, including risk assessment, fraud detection, and medical diagnosis (Zheng et al., 2022). Unlike Balla et al who used CNN-LSTM models for the experiment, there is need to experiment on various algorithms for various tasks to establish the true impact of imbalanced datasets on ML model performance.

Research on feature scaling (e.g., normalization, standardization) by Prakash (2024) has demonstrated its critical importance in algorithms like Random Forest (RF), Decision Trees (DT), Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN) that are sensitive to feature magnitudes. In health, the accuracy of the DT, RF and Regression models trained using feature engineered datasets to predict a binary classification task related to the existence of a heart attack is conducted. The results demonstrate the model's performance improved significantly with DT classifier exhibiting exceptional results. This experiment also established that improper use of feature engineering combinations has a negative impact on model accuracy. Additionally, the importance of feature selection and dimensionality reduction methods, such as Principal Component Analysis (PCA) and feature importance from tree-based models, might increase model efficiency and avoid overfitting, especially in high-dimensional datasets (BÜYÜKKEÇECİ & Okur, 2022). The dataset used for this experiment is contained only 303 instances and 75 attributes which despite having a wide range of algorithms, may not be adequate for training a ML model. This study uses more dataset instances to enhance the performances of the tested algorithms.

B. Examination of Common Data Preprocessing Techniques

Contemporary applications generate vast amounts of data, often containing irrelevant information (Amato & Di Lecce, 2024). ML models, whose performance is influenced by the quality of data, extract knowledge from these datasets (Borodkin et al., 2023). Data preprocessing is a crucial process for overcoming data quality issues such as noisy, redundant, or missing data values (Fan et al., 2021). Data preprocessing involves evaluating data quality, synchronizing and integrating data, cleaning noisy data, reducing high dimensionality, and transforming inappropriate data formats. These activities improve the quality of data, with each activity having a positive impact on the data used for ML model development (Frye et al., 2021). Figure 1 summarizes the common data preprocessing techniques that will be examined in this study.

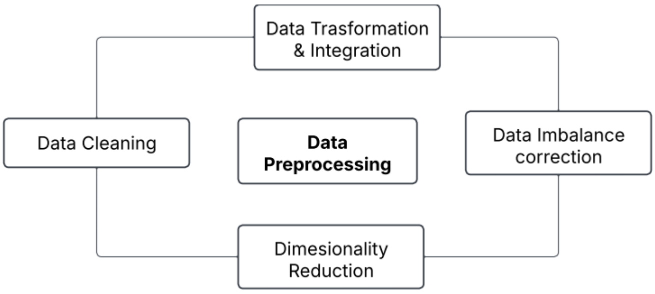

Figure1: Data preprocessing techniques
Source: (Author, 2025)

i. Data Cleaning

The fact that we will rarely come across flawlessly prepared and clean data is an important aspect to acknowledge as we delve into the realm of data science. The majority of the time, our initial datasets will have a number of quality problems. Some of these data quality issues that should be checked include missing values, outliers, and inconsistent formatting (Data Science Horizons, 2023). Missing values are the ghosts in data science due to human error, data gathering issues, or irrelevant data fields. These issues can introduce bias or skewed analyses, requiring appropriate treatment to maintain research quality (Brijith, 2023).

Data cleaning is also applied due to outliers which are comparable the data's "black sheep." These observations differ significantly when compared to other data points within a dataset. They may be the result of legitimate but exaggerated observations, measurement errors, or data input problems. These data may have drastic impacts on the results, either exaggerating the results or reducing the scores significantly (Data Science Horizons, 2023).

In a perfect world, all data would have a standard format, which would greatly simplify the work of data scientists. Regretfully, it is rarely the case. Human error, system modifications, or the combining of data from many sources can all result in inconsistent data formatting, a typical quality problem. These discrepancies may appear in date formats, string data casing, or text-based numeric data, among other formats (Frye et al., 2021). Data cleaning techniques improve data quality by ensuring that the dataset is complete, accurate, and uniformly formatted because knowing how good your data is can have a big impact on your studies, from the conclusions you make to how accurate your prediction models are (Data Science Horizons, 2023). Data cleaning can be carried either automatically by a computer program or manually using data wrangling tools (Lee et al., 2021).

ii. Data Transformation

One of the most important steps in the ML workflow is data transformation. It entails transforming unprocessed data into a format that is better suited for analysis and training ML models (Fan et al., 2021). When it comes to data preprocessing, especially data transformation, ML engineers must exercise caution. To do this, the data must be formatted to make analysis easier. Typical methods for transforming data include normalization, standardization, and discretization. While normalization scales the data to a uniform range, standardization modifies the data to have a variance of one and a mean of zero. To divide continuous data into distinct categories, discretization is utilized (Prakash, 2024).

Data for ML model training come in many forms and in most circumstances, the data points in different columns values may not be on similar scale and may possess extremely low and extremely high values in magnitude. In such cases column data normalization or standardization is used to bring them on to similar scale. This is called feature scaling (Strasser & Klettke, 2024). This will help in faster convergence of ML models. Normalization is a feature scaling method in which values are adjusted so that they range between 0 and 1. This is also termed as Min-Max scaling (Frye et al., 2021).

iii. Data Imbalance Correction (DIC)

Most challenges in the real world inevitably involve imbalanced data. This results to the minority class often being dismissed as noise especially when the minority-to-majority ratio, or imbalance ratio (IR), is low. As a result, the ML model exhibits bias towards the majority class, resulting in a higher number of

False Positives (FP) and a lower number of True Positives (TP) (Werner de Vargas et al., 2023). This issue of dataset bias can be addressed in a number of ways, such as by placing more attention on under-represented data samples (Jones et al., 2023). Eqn 1 is used to determine the imbalance ratio of a dataset.

$$Imbalance\ Ratio\ (IR) = \frac{Minority\ Instances}{Majority\ Instance}$$

The result is usually between 0-1 and is interpreted as shown in Table 1.

TABLE I INTERPRETATION OF THE DEGREE OF IMBALANCE

Degree of Imbalance	**Minority class proportion**
Mild	21-40% of the dataset
Moderate	1-20% of the dataset
Extreme	<1% of the dataset

Correcting data imbalance can be done using algorithms to either oversample the minority sample, under-sample the majority sample, or both (hybrid sampling) before learning. It can also be accomplished through algorithmic processes: where processing learning use algorithms, like cost- sensitive and ensemble ML models,that are optimized for unbalanced data (Al-Mhiqani et al., 2021b).

iv. Feature Selection/Dimensionality Reduction

In ML, feature selection is an essential procedure that is frequently mentioned as having a greater influence on ML models' performance than algorithm selection (BÜYÜKKEÇECİ & Okur, 2022). Feature engineering enhances the functionality of ML models by choosing, converting, and producing features from unprocessed data. The model's capacity to learn and generalize from the data is directly impacted by the quality of the selected features (Zheng & Casari, 2018). Effective feature engineering, as noted by Zhang et al., (2022), can result in notable increases in model accuracy, occasionally even outperforming the benefits of hyperparameter adjustment.

The significance of feature engineering is due to a number of important considerations. First, well selected features reveal the underlying relationships and patterns in the data, which improves prediction accuracy. This is because ML algorithms learn from the features presented to them, and if those features are not informative or relevant, the model's performance will be limited (Ailyn, 2024). Secondly, model interpretability can be improved through feature engineering. It is simpler to comprehend how the model generates its predictions when features are developed that have a distinct and significant relationship to the target variable. This is especially significant in fields like healthcare and finance where explainability is essential (BÜYÜKKEÇECİ & Okur, 2022).

According to Zhang et al.,(2022), the three primary issues that feature selection tackles are noise, over-fitting, and dimensional catastrophe. By using high-quality features, feature selection can not only lower computation and model complexity but also enhance the model's final prediction. Feature selection can be done using three techniques: filtering, wrapping, and embedding. Despite the emergence of automated feature engineering techniques (e.g., Deep Learning (DL) models), human skill and domain knowledge continue to play a critical role. For features to be meaningful and useful, it is necessary to comprehend the issue domain and the underlying data. To ascertain which features are best for the model, feature engineering necessitates testing and assessment because it is frequently an iterative process (Lecun et al., 2015). To evaluate the effect of feature engineering on prediction accuracy, models with and without engineered features are compared. Metrics like R-squared, Mean Squared Error (MSE), and Mean Absolute Error (MAE) are used to assess performance gains and show how feature engineering approaches work (Ailyn, 2024).

Data cleaning, transformation, imbalance correction and feature engineering are common data preprocessing techniques employed to improve data quality for ML models training and validation. According to Fan et al., (2021), more than 80% of ML model development is taken by data purification step, hence, it is crucial to determine which of the techniques has a high impact on the ML models' performance so that developers can select what is useful to them during development so as to minimize on time wastage.

METHODS AND MATERIAL

The purpose of this experiment is to investigate how various data preprocessing techniques affect the performance of ML models. This study hypothesizes that the four data preprocessing techniques (Cleaning, transformation, imbalance correction and feature engineering) improves quality of data which as a result has a positive impact on ML models' performance. Another hypothesis is that of the four preprocessing techniques, some have higher positive impact on ML models' performance than others. This experiment will consider four (4) datasets and four (4) ML algorithms.

A. The Datasets

Four (4) public datasets downloadable from Kaggle have been used for this study. The datasets are as summarized in Table 2. These datasets having varied formats are meant for training various ML classification and regression models. The selection of these datasets was based on their representation of real-world situations and have been frequently used in many experiments.

TABLE II COMMON PUBLIC DATASETS FOR THE STUDY

Dataset	Category	Statistics	Features
Enron (Federal Energy Regulatory Commission. (2004)	Insider threat detection	517401 records of emails with two (2) features from 150 employees	– Email logs
CERT r2 (CMUSEI, 2016)	Insider threat detection	34190166 records with 41 features collected from over 4000 employees for 18 months	– Email logs – Authentication logs – Web browsing activities – USB device use logs – File access logs
CERT r4.2 (Lindauer, (2020)	Insider threat detection	32,770,227 records with 92 features collected from over 1000 employees for 18 months	– Email logs – Authentication logs – Web browsing activities – USB device use logs – File access logs
Student performance (Cortez & Silva, 2008)	Education analytics	Over 20000 records of 1314 students with 33 features collected over one academic year.	– Demographics – Social factors – Academic history – Behavioral aspects

B. Machine Learning Algorithms

Different ML models exhibit sensitivity to unique data pre-processing activities. Based on this theory Borodkin et al., (2023) presents four ML algorithms classifications: (a) Those sensitive to feature engineering; (b) Algorithms sensitive to feature scaling and normalization; (c) algorithms sensitive to outliers and noise, and (d) algorithms sensitive to data imbalance. Four ML algorithms were selected from the four groups to demonstrate the impact of data preprocessing on ML algorithms performance.

i. **Random Forest**: This is an ensemble learning technique for classification and regression problems. It generates either the majority class (for classification) or the average prediction (for regression) after building several Decision Trees (DTs) during training. In RF, several DTs are combined to decrease overfitting and enhance the algorithm's accuracy (Salman et al., 2024). In addition, each tree is trained individually using random subsets of data using Bagging. To maintain diversity among trees, a random subset of features is taken into account at each split. Despite the advantages, RF can be computationally expensive when many features are introduced hence the need for preprocessing (Boyko et al., 2022).

ii. **Logistic Regression:** LR is a binary classification statistical technique whose objective is to forecast one of two potential outcomes. It uses the sigmoid function to get an estimate of the likelihood that a given input is a member of a specific class (Yun, 2021). Suitable for probability estimation, this function converts any real-valued number to a range between 0 and 1. Logistic regression's ease of use, interpretability, and efficiency in handling linearly separable data make it popular in a variety of domains, such as text classification, fraud detection, and medical diagnosis (Portl, 2021).

iii. **K-Nearest Neighbors (k-NN):** This is non-parametric supervised learning technique for regression and classification. It determines the k data points that are the nearest to a given input, or neighbors, and then predicts the results based on the average value (for regression) or the majority class (for classification). KNN uses distance measures, such as Euclidean distance, to calculate how similar two data points are. It is good for small to medium-sized datasets but experience high computational costs as the size of the dataset increases (Zhang, 2016).

iv. **Long Short Term Memory (LSTM)**: This is a RNN type that efficiently learns and remembers long-term dependencies in sequential data. The information flow is managed by a gated architecture that includes input, output, and forget gates. This makes them ideal for tasks like anomaly detection, machine translation, and speech recognition since gates enable it to identify patterns in time-series data. LSTMs can describe complicated temporal dependencies because of their capacity to retain historical information over long sequences (Houdt et al., 2020).

C. Hyperparameter Settings and Evaluation Metrics

Baseline hyperparameters used for the experiment as indicated in Table 3.

TABLE III BASELINE HYPERPARAMETERS

Hyperparameter	Hyperparameter settings	Remark
Batch size	256	To utilize GPU power
Validation split	0.2	80% used for training
Patience	5	# of epochs set to terminate model training after convergence.
Number of	25	# of times the entire dataset is passed through the model during

Hyperparameter	Hyperparameter settings	Remark
epochs		training
Learning rate	0.001	The pace at which the model learns

To illustrate the selected models' performance, four evaluation metrics which include Accuracy, Precision, Recall and F-measure were used. The equations for the metrics are as indicated below.

i. $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$

ii. $Precision = \frac{TP}{TP+FP}$

iii. $Recall = \frac{TP}{TP+FN}$

iv. $F_1\ score = \frac{2*Precision*Recall}{Precision+Recall}$

Accuracy refers to the number of correctly classified data instances divided by the total number of data instances (Gong, 2021) while Precision is the number of true positives divided by the total number of positive predictions. Precision is simply the number of correctly predicted cases and its level should be as high as possible. A value of 0.0 means no Precision, while 1.0 is perfect (Choudhary, 2020). Recall determines the number of actual positive cases that the model correctly predicts. Recall gives values between 0.0-1.0. 0.0 means no Recall, while 1.0 means complete or perfect recall. F_1 Score evaluates a model's predictive abilities by analyzing its performance on each class independently rather than taking into account overall performance, as accuracy does. F1score is considered perfect when its value is 1 and if the value is less than 0.5, it means the classifier has more FPs (Jiang & Luo, 2022).

RESULTS AND DISCUSSION

Results for the performance are displayed according to the four selected datasets.

A. Enron Dataset

TABLE IV ENRON DATASET ANALYSIS RESULTS

ML Algorithm	Data preprocessing technique	Evaluation Metrics (%)			
		Accuracy	Recall	Precision	F1-Score
Logistic Regression	Baseline	80.50	25.00	85.00	38.46
	Cleaning	81.00	26.00	86.00	40.00
	Transformation	82.00	28.00	87.50	42.42
	Feature Selection	83.00	30.00	88.00	44.44
	DIC	82.50	27.50	87.00	41.79
Random Forest	Baseline	85.00	32.00	89.00	47.06
	Cleaning	85.50	33.00	90.00	48.53
	Transformation	87.00	35.00	91.50	50.72
	Feature Selection	88.00	37.00	92.00	52.86
	DIC	86.50	34.50	90.50	50.00
LSTM	Baseline	78.00	20.00	80.00	32.00
	Cleaning	78.50	21.00	81.00	33.33
	Transformation	80.00	23.00	82.50	36.07
	Feature Selection	81.00	25.00	83.00	38.46
	DIC	80.50	24.00	82.00	37.21

ML Algorithm	**Data preprocessing technique**	**Evaluation Metrics (%)**			
		Accuracy	**Recall**	**Precision**	**F1-Score**
K-NN	Baseline	81.50	27.00	86.50	41.22
	Cleaning	82.00	28.00	87.00	42.42
	Transformation	84.00	31.00	89.00	45.93
	Feature Selection	85.00	32.50	90.00	47.62
	DIC	83.50	30.00	88.50	44.78

B. CERT r2 Dataset

TABLE V CERT r2 DATASET ANALYSIS RESULTS

ML Algorithm	**Data preprocessing technique**	**Evaluation Metrics (%)**			
		Accuracy	**Recall**	**Precision**	**F1-Score**
Logistic Regression	Baseline	82.4	82.4	82.0	82.2
	Cleaning	82.4	82.4	82.0	82.2
	Transformation	83.2	83.2	83.1	83.3
	Feature Selection	83.3	83.3	83.0	83.2
	DIC	83.7	83.7	83.2	83.5
Random Forest	Baseline	82.4	82.4	82.0	82.0
	Cleaning	82.4	82.4	82.0	82.0
	Transformation	83.5	83.5	83.0	83.2
	Feature Selection	84.0	84.0	84.1	81.3
	DIC	85.0	85.0	84.5	84.7
LSTM	Baseline	81.0	81.0	80.5	80.7
	Cleaning	81.0	81.0	80.5	80.7
	Transformation	82.1	82.1	82.5	82.7
	Feature Selection	82.5	82.5	81.5	81.7
	DIC	84.0	84.0	83.5	83.7
K-NN	Baseline	78.0	78.0	77.5	77.7
	Cleaning	78.0	78.0	77.5	77.7
	Transformation	79.5	79.5	79.0	79.2
	Feature Selection	81.0	81.0	80.5	80.7
	DIC	83.5	83.5	83.0	83.2

C. The CERT r4.2 Dataset

TABLE VI CERT r4.2 DATASET ANALYSIS RESULTS

ML Algorithm	**Data preprocessing technique**	**Evaluation Metrics (%)**			
		Accuracy	**Recall**	**Precision**	**F1-Score**
Logistic Regression	Baseline	85.0	85.0	84.5	84.7
	Cleaning	85.0	85.0	84.5	84.7
	Transformation	86.0	86.0	85.5	85.7

ML Algorithm	Data preprocessing technique	Evaluation Metrics (%)			
		Accuracy	Recall	Precision	F1-Score
	Feature Selection	87.0	87.0	86.5	86.7
	DIC	88.0	88.0	87.4	87.6
Random Forest	Baseline	89.0	89.0	88.5	88.7
	Cleaning	89.0	89.0	88.5	88.5
	Transformation	88.5	88.5	88.0	88.2
	Feature Selection	87.5	87.5	87.0	87.2
	DIC	89.0	89.0	89.5	89.7
LSTM	Baseline	81.5	81.5	81.0	81.2
	Cleaning	81.5	81.5	81.0	81.2
	Transformation	82.0	82.0	81.5	81.7
	Feature Selection	82.5	82.5	82.0	82.2
	DIC	84.5	84.5	84.0	84.2
K-NN	Baseline	78.0	78.0	77.5	77.7
	Cleaning	78.0	78.0	77.5	77.7
	Transformation	79.5	79.5	79.0	79.2
	Feature Selection	81.0	81.0	80.5	80.7
	DIC	83.5	83.5	83.0	83.2

D. The Student Performance dataset

TABLE VII STUDENT PERFORMANCE DATASET ANALYSIS RESULTS

ML Algorithm	Data preprocessing technique	Evaluation Metrics (%)			
		Accuracy	Recall	Precision	F1-Score
Logistic Regression	Baseline	91.15	94.65	94.20	94.43
	Cleaning	91.15	94.65	94.20	94.43
	Transformation	91.45	95.61	93.73	94.66
	Feature Selection	91.00	94.46	94.19	94.33
	DIC	90.09	93.31	94.12	93.72
Random Forest	Baseline	91.45	97.23	92.38	94.74
	Cleaning	91.68	97.23	92.63	94.87
	Transformation	91.98	97.52	92.73	95.07
	Feature Selection	91.75	96.37	93.43	94.88
	DIC	90.32	94.94	92.98	93.95
LSTM	Baseline	87.14	90.93	92.70	91.80
	Cleaning	86.91	90.83	92.51	91.66
	Transformation	81.77	94.36	84.44	89.13
	Feature Selection	88.35	92.36	92.89	92.62
	DIC	87.37	91.12	92.80	91.95
K-NN	Baseline	84.72	97.90	85.06	91.03

ML Algorithm	Data preprocessing technique	Evaluation Metrics (%)			
		Accuracy	Recall	Precision	F1-Score
	Cleaning	85.10	92.45	89.13	90.76
	Transformation	91.38	96.85	92.60	94.68
	Feature Selection	82.75	98.57	82.89	90.05
	DIC	89.18	91.40	94.75	93.05

The performance of ML models is significantly impacted by data pre-processing, as demonstrated by this study, which also shows a distinct hierarchy of effectiveness across various approaches. Our findings show that feature selection consistently yields the most performance gains across four datasets and four assessment metrics: the Student Performance dataset, CERT R4.2, CERT R2, and Enron. Model accuracy and efficiency are increased, overfitting is decreased, and generalization is enhanced by eliminating superfluous or unnecessary features implying a crucial process that should be done to all datasets during ML model development.

Data transformation follows feature selection closely proving to be the next most effective technique. For the student performance dataset, the values across the four metrics for the data transformation process pushes the accuracy values of the models above 90% for three instances. These values within this dataset outdo all other data preprocessing procedures. This implies that model convergence is improved by normalization and standardization, especially for algorithms that are sensitive to feature scales. This leads to better performance in regression and classification.

Data imbalance correction (DIC) comes in third, especially for datasets like CERT R2 and R4.2 that have skewed class distributions as shown in Table 2 and Table 3. For example, in Table 2 of the CERT r4.2 dataset, the accuracy of the LR (83.7), RF(85), LSTM (84) and KNN (83.3) are the highest among other preprocessing tasks. This signifies that DIC is an important step to be performed on classification datasets. Although weighting and resampling procedures increase F1-score and Recall, their effects vary depending on the degree of imbalance and are dataset-dependent. DIC is crucial for datasets used for classification tasks and especially it has a big impact on binary classification. If the dataset is bias, the minority group will be ignored which will result in to skewed predictions towards the majority group.

Apart from the student performance dataset's results, data cleaning for three other datasets did not have a significant impact on the model's performance. For example, in the Enron's dataset, it yielded 81% which is the same figure as baseline accuracy, CERT r2 gave 82.4 while CERT r4.2 gave 85.1% as the baseline. This implies that data cleaning had the least effect on the accuracy of all ML model performance. Although dealing with outliers and missing values enhances data integrity, its impact is minimal unless the dataset contains notable irregularities.

CONCLUSION

This study illustrates how important data preprocessing is to improving ML model performance. Through rigorous examination, it was discovered that feature engineering significantly impacted model accuracy and generalization more than any other preprocessing methodology. This demonstrates how crucial it is to pick and build pertinent features with care in order to accurately depict the underlying patterns in the data. Following feature engineering is data transformation which encompass normalization, scaling, and encoding. This shows the need to critically make sure that data for ML model development is in an appropriate format and range.

The study also established that data imbalance correction techniques such class weighting and resampling were useful in enhancing model performance and fairness, especially when dealing with skewed class distributions. It is noteworthy that, in contrast to the other methods, data cleansing had the least effect, although being crucial for guaranteeing data quality. This may imply that the advantages of engineering and changing features outweigh the profits from additional cleaning once fundamental data integrity has been guaranteed.

Overall, the results highlight a focused approach to preprocessing, with feature engineering and transformation being the primary focus, followed by data imbalance resolution and basic data cleaning. Such a strategy can result in far better ML models and more effective use of resources and time. Although this study offers a clear understanding of the relative effects of different data preprocessing methods, we recommend an evaluation of each of the various methods of accomplishing the preprocessing tasks to enable the users to settle for more efficient and effective ones.

References

[1]. Ailyn, D. (2024). (PDF) Feature Engineering for Financial Market Prediction: From Historical Data to Actionable Insights. https://www.researchgate.net/publication/383908810_Feature_Engineering_for_Financial_Market_Prediction_From_Historical_Data_to_Actionable_Insights

[2]. Amato, A., & Di Lecce, V. (2024). (PDF) Data preprocessing impact on machine learning algorithm performance. ResearchGate. https://doi.org/10.1515/comp-2022-0278

[3]. Angelovič, M., Krištof, K., Jobbágy, J., & Findura, P. (2018). (PDF) The effect of conditions and storage time on course of moisture and temperature of maize grains. BIO Web of Conferences. https://doi.org/10.1051/bioconf/20181002001

[4]. Balla, A., Habaebi, M. H., Elsheikh, E. A. A., Islam, M. R., & Suliman, F. M. (2023). The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems. Sensors, 23(2), Article 2. https://doi.org/10.3390/s23020758

[5]. Borodkin, K., Nurtas, M., Altaibek, A., Daineko, Y., & Otepov, T. (2023). Data Pre-processing and Visualization for Machine Learning Models and its Applications in Education. 8th International Conference on Digital Technologies in Education, Science and Industry.

[6]. Boyko, N., Omeliukh, R., & Duliaba, N. (2022). The Random Forest Algorithm as an Element of Statistical Learning for Disease Prediction.

[7]. Brijith, A. (2023). (PDF) Data Preprocessing for Machine Learning. ResearchGate. https://www.researchgate.net/publication/375003512_Data_Preprocessing_for_Machine_Learning

[8]. BÜYÜKKEÇECİ, M., & Okur, M. (2022). (PDF) A Comprehensive Review of Feature Selection and Feature Selection Stability in Machine Learning. ResearchGate. https://doi.org/10.35378/gujs.993763

[9]. Dandu, M. M. K., Jain, J., Vijayabaskar, S., & Goel, P. (2024). Assessing the Impact of Data Imbalance on the Predictive Performance of Machine Learning Models | Request PDF. ResearchGate. https://doi.org/10.1109/IC3I61595.2024.10829313

[10]. Data Science Horizons. (2023). Data Cleaning and Preprocessing for Data Science Beginners. Data Science Horizons.

[11]. Fan, C., Chen, M., Wang, X., Wang, J., & Bufu, H. (2021). (PDF) A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building

Operational Data. ResearchGate. https://doi.org/10.3389/fenrg.2021.652801

[12]. Frye, M., Mohren, J., & Schmitt, R. H. (2021). Benchmarking of Data Preprocessing Methods for Machine LearningApplications in Production. 54th CIRP Conference on Manufacturing Systems. https://doi.org/doi.org/10.1016/j.procir.2021.11.009

[13]. Houdt, G. V., Carlos Mosquera, & Nápoles, G. (2020). (PDF) A Review on the Long Short-Term Memory Model. https://doi.org/DOI:10.1007/s10462-020-09838-1

[14]. Jamshed, H., Khan, M. S. A., Khurram, M., Inayatullah, S., & Athar, S. (2019). Data Preprocessing: A preliminary step for web data mining. 3C Tecnología_Glosas de Innovación Aplicadas a La Pyme, 206–221. https://doi.org/10.17993/3ctecno.2019.specialissue2.206-221

[15]. Jones, H. R., Mu, T., Andrei C., P., & Yusuf, S. (2023). (PDF) Adapting Data-Driven Techniques to Improve Surrogate Machine Learning Model Performance. ResearchGate. https://doi.org/10.1109/ACCESS.2023.3253429

[16]. Koresh, H. J. D. (2024). Impact of the Preprocessing Steps in Deep Learning-Based Image Classifications. National Academy Science Letters, 47(6), 645–647. https://doi.org/10.1007/s40009-023-01372-2

[17]. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[18]. Lee, G. Y., Alzamil, L., Doskenov, B., & Termehchy, A. (2021). (PDF) A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance. https://doi.org/DOI:10.48550/arXiv.2109.07127

[19]. Li, P., Rao, X., Blase, J., Zhang, Y., Chu, X., & Zhang, C. (2021). CleanML: A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks. 2021 IEEE 37th International Conference on Data Engineering (ICDE), 13–24. 2021 IEEE 37th International Conference on Data Engineering (ICDE). https://doi.org/10.1109/ICDE51399.2021.00009

[20]. Liew, Y. C., Chuan, Y., Lim, T. Y., Tan, C. J., Chai, K. K., & Deng, X. (2024). The Effect of Data Transformation Techniques on Machine Learning Performance: A Case Study on Student Dropout Prediction | IEEE Conference Publication | IEEE Xplore. https://doi.org/DOI: 10.1109/PRML62565.2024.10779714

[21]. Portl, S. U. (2021). Logistic Regression. In Categorical Data Analysis. Newsom.

[22]. Prakash, Dr. A. A. (2024). Pre-processing techniques for preparing clean and high-quality data for diabetes prediction. International Journal of Research Publication and Reviews, 5(2), 458–465. https://doi.org/10.55248/gengpi.5.0224.0412

[23]. Salman, H. A., Kalakech, A., & Steiti, A. (2024). (PDF) Random Forest Algorithm Overview. https://doi.org/DOI:10.58496/BJML/2024/007

[24]. Strasser, S., & Klettke, M. (2024). Transparent Data Preprocessing for Machine Learning. Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics, 1–6. https://doi.org/10.1145/3665939.3665960

[25]. Wanyonyi, E. N., Masinde, N. W., & Abeka, S. O. (2024). A Theory-Based Deep Learning Approach for Insider Threat Detection and Classification. International Journal of Computer Applications Technology and Research. https://doi.org/10.7753/IJCATR1310.1004

[26]. Werner de Vargas, V., Schneider Aranda, J. A., dos Santos Costa, R., da Silva Pereira, P. R., & Victória Barbosa, J. L. (2023). Imbalanced data preprocessing techniques for machine learning: A systematic mapping study. Knowledge and Information Systems, 65(1), 31–57. https://doi.org/10.1007/s10115-022-01772-8

[27]. Zhang, M., Lu, J., Ma, N., Cheng, T. C. E., & Hua, G. (2022). (PDF) A Feature Engineering and Ensemble Learning Based Approach for Repeated Buyers Prediction. ResearchGate. https://doi.org/10.15837/ijccc.2022.6.4988

[28]. Zhang, Z. (2016). (PDF) Introduction to machine learning: K-nearest neighbors. ResearchGate. https://doi.org/10.21037/atm.2016.03.37

[29]. Zhao, Y., Huang, Z., Gong, L., & Zhu, Y. (2023). (PDF) Evaluating the Impact of Data Transformation Techniques on the Performance and Interpretability of Software Defect Prediction Models. https://doi.org/DOI:10.1049/2023/6293074

[30]. Zheng, A., & Casari, A. (2018). Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists | Guide books | ACM Digital Library. O'Reilly Media, Inc. https://dl.acm.org/doi/10.5555/3239815

[31]. Zheng, M., Wang, F., Hu, X., Miao, Y., Cao, H., & Tang, M. (2022). A Method for Analyzing the Performance Impact of Imbalanced Binary Data on Machine Learning Models. Axioms, 11(11), Article 11. https://doi.org/10.3390/axioms11110607