engineering proceedings

MDPI

Proceeding Paper

Effect of Data Preparation on Machine Learning Models for Diabetes Prediction †

Goran Martinović [1,‡], Ivan Ivković [1,*], Domen Verber [2] and Tatjana Bačun [3,4,5]

[1] Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Kneza Trpimira 2B, 31000 Osijek, Croatia; goran.martinovic@ferit.hr
[2] Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška Cesta 46, 2000 Maribor, Slovenia; domen.verber@um.si
[3] Division of Endocrinology, Department of Internal Medicine, University Hospital Centre Osijek, Josipa Huttlera 4, 31000 Osijek, Croatia; tatjana.bacun@mefos.hr
[4] Department of Internal Medicine and History of Medicine, Faculty of Medicine Osijek, Josip Juraj Strossmayer University of Osijek, Josipa Huttlera 4, 31000 Osijek, Croatia
[5] Department of Clinical Medicine, Faculty of Dental Medicine and Health Osijek, Josip Juraj Strossmayer University of Osijek, Crkvena Ulica 21, 31000 Osijek, Croatia
* Correspondence: ivan.ivkovic@ferit.hr
† Presented at the 34th International Scientific Conference on Organization and Technology of Maintenance (OTO 2025), Osijek, Croatia, 12 December 2025.
‡ Deceased author.

Abstract

This paper examines how data preparation affects machine-learning classifiers for diabetes-risk prediction using the Pima Indians Diabetes Database. Three preprocessing methods are considered: imputing invalid zeros, handling outliers, and data scaling. Nine algorithms are evaluated on this dataset: linear/probabilistic baselines (Logistic Regression, Gaussian Naive Bayes), distance-based methods (KNN, Support Vector Machines), a single tree-based model (Decision Tree), and tree ensembles (Random Forest, Gradient Boosting, XGBClassifier, LightGBM). Median imputation of invalid zeros yields the largest and most consistent gains in accuracy and AUC. Outlier handling uses interquartile-range filtering, with Local Outlier Factor as an auxiliary indicator; effects are modest for accuracy and small, model-dependent for AUC. Scaling offers targeted benefits: for KNN, robust scaling can slightly alter performance and may reduce AUC relative to median-only imputation in this setup; SVM shows modest gains, while tree ensembles are comparatively insensitive overall. Ensembles achieve the highest performance and remain robust under minimal preparation, while simpler models benefit most from pipelines combining median imputation, careful outlier handling, and appropriate scaling. Hyperparameter tuning yields small to substantial gains—large for Decision Trees—while leaving ensemble rankings largely unchanged. Overall, results highlight the centrality of median imputation and the selective value of scaling for distance-based classifiers in diabetes-risk prediction.

Keywords: machine learning; data preparation; diabetes; classification

Check for updates

Academic Editor: Časlav Livada, Mirko Karakašić, Naida Ademović, Tomislav Keser, Ercan Işık and Silva Lozančić

Published: 28 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Diabetes mellitus is among the most prevalent chronic endocrine diseases worldwide and continues to rise in incidence, increasingly affecting younger populations as lifestyles change [1]. It is characterized by persistent hyperglycaemia arising from insufficient insulin secretion and/or reduced insulin sensitivity, which progressively disrupts carbohydrate, fat, and protein metabolism [2]. The clinical course involves a long asymptomatic

phase, while complications span microvascular (retinopathy, nephropathy, neuropathy) and macrovascular outcomes that elevate morbidity and mortality. For risk assessment and early recognition, salient factors include age, adiposity and fat distribution (e.g., BMI, waist circumference), blood pressure, lipid profile (LDL/HDL), physical inactivity, dietary patterns, smoking, family history, and prior gestational diabetes [3]. Classic symptoms such as polyuria, polydipsia, polyphagia, fatigue, blurred vision, and unexplained weight loss may be present but are not necessary for diagnosis.

Machine learning (ML) methods increasingly support early, scalable risk stratification by exploiting structured clinical features from routine measurements and questionnaires [4]. A broad repertoire of classifiers—linear and probabilistic models, distance-based methods, decision trees, and tree ensembles—enables accurate prediction when models are well specified. However, real-world datasets often contain "noise": medically invalid zeros or missing values, outliers, and heterogeneous feature scales. If left unaddressed, these artifacts bias parameter estimation, distort decision boundaries, and may inflate apparent performance through data leakage or unstable thresholds. Consequently, principled data preparation is an essential prerequisite for credible ML-based screening and decision support [5].

This paper quantifies the effect of three foundational preparation steps—imputation of missing/invalid entries, outlier handling, and data scaling—affect downstream classifier performance on the widely used Pima Indians Diabetes Database. The evaluation spans representative algorithms: Logistic Regression and Gaussian Naive Bayes; KNN and Support Vector Machines; Decision Tree and ensemble methods including Random Forest, Gradient Boosting, LightGBM, and XGBClassifier.

This paper is organized as follows. Section 2 details the methodology, including the dataset and its characteristics, data preparation (imputation of missing values, outlier handling, and data scaling), and the classification algorithms. Section 3 presents the results and their analysis.

2. Methodology

In this section, the dataset used, the data preparation methods, and the classification algorithms are described.

2.1. Dataset and Characteristics

The Pima Indians Diabetes Database [6] was selected for its realistic data-quality challenges—missing values, class imbalance, heterogeneous feature scales, and outliers. It contains 768 samples with eight numerical features (Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, DiabetesPedigreeFunction, Age) and a binary classification target (diabetes: yes/no). Class distribution includes 268 positive cases (35%) and 500 negative cases (65%), representing a 1:1.87 imbalance common in healthcare applications. Approximately 7% of feature values are encoded as zero but semantically indicate missing data. Feature scales are heterogeneous: Age ranges 21–81 years, Glucose 0–199 mg/dL, and Insulin 0–846 μU/mL. This combination of challenges—missingness, class imbalance, outliers, and scale heterogeneity—reflects realistic healthcare data characteristics, justifying the dataset's selection for this study.

2.2. Data Preparation Methods

Figure 1 shows the workflow of the research, which consists of dataset loading, exploratory data analysis, missing value imputation, outlier handling, data scaling, models training, hyperparameter tuning in the final phase, and models evaluation.

https://doi.org/10.3390/engproc2026125013

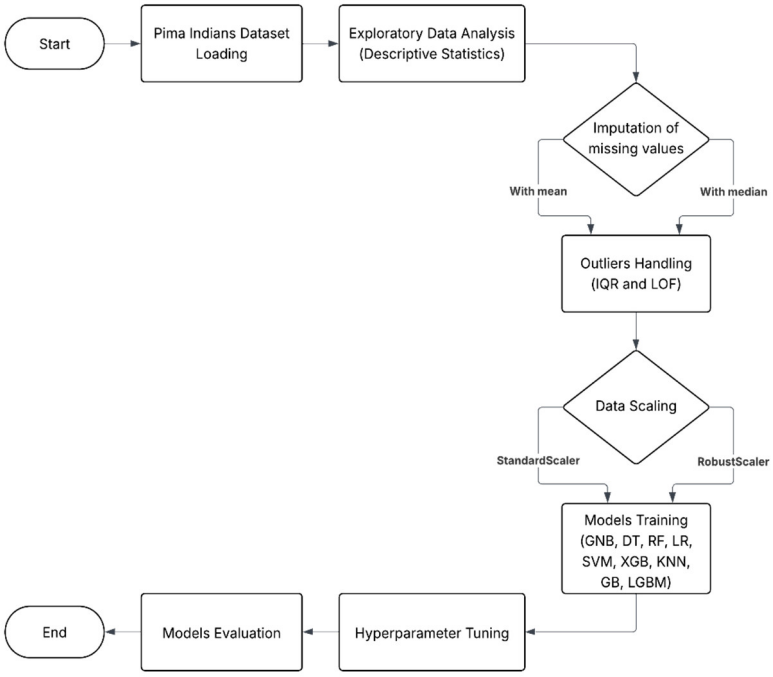


Figure 1. Research Workflow.

2.2.1. Missing Value Imputation

Two imputation strategies were tested. Median imputation replaces missing values with feature median, chosen over mean due to robustness against outliers particularly important in healthcare data [7]. Mean imputation was tested as alternative baseline. Both strategies represent computationally efficient approaches implementable in standard ML frameworks. Imputation parameters were computed on training fold only and applied to test fold, preventing data leakage. More sophisticated techniques (MICE, KNN imputation) were excluded to focus on widely used practical approaches deployable in resource-constrained settings.

2.2.2. Outlier Handling

Two outlier handling methods were applied: Interquartile Range (IQR) and Local Outlier Factor (LOF). Interquartile Range (IQR) method removed samples with feature values beyond $[Q_1 - k{\cdot}IQR,\ Q_3 + k{\cdot}IQR]$, with a factor k, usually set to 1.5. This interpretable, distribution-free approach with standard $1.5{\cdot}IQR$ multiplier removes approximately 2–3% of samples. IQR-based handling was chosen because it is widely understood, implementable in any framework, and represents practical baseline approach rather than sophisticated statistical method [8]. LOF (Local Outlier Factor) is an unsupervised algorithm that identifies outliers based on local neighboring points and is based on a single parameter, the minimum number of nearest neighbors, which is used to calculate the factor itself. It quantitatively assesses how exceptional an object is [9]. A LOF value close to 1 indicates that the object is not an outlier, while higher values reflect the corresponding degree of anomaly.

2.2.3. Data Scaling

Two scaling approaches were compared: StandardScaler and RobustScaler. StandardScaler centers distributions at 0 with unit variance, and it is calculated by Equation (1):

$$z = \frac{x - \mu}{\sigma}, \tag{1}$$

where z is the scaled value, x is the original value, μ is the mean of all feature values, and σ is the standard deviation of all feature values. RobustScaler, more robust to extreme values, is calculated by Equation (2):

$$z = \frac{x - Q_1(x)}{Q_3(x) - Q_1(x)}, \quad (2)$$

where z is the scaled value, x is the original value, $Q_1\ (x)$ and $Q_3\ (x)$ are the first (25th percentile) and third (75th percentile) quartiles of the feature, respectively [10]. Each quartile is a single statistical value computed from the entire distribution of the feature across all samples, representing the points below which 25% and 75% of the data lie. It was compared to investigate whether outlier robustness justifies increased complexity. Scaling parameters were computed on training fold only, preventing data leakage.

2.3. Classification Algorithms

Nine algorithms were selected to cover complementary inductive biases, interpretability levels, and sensitivity to preprocessing, forming a diverse yet tractable set of foundational baselines suitable for side-by-side comparison. Simple classifiers included Logistic Regression (linear, interpretable, scale-sensitive), Gaussian Naive Bayes (probabilistic, distribution-sensitive), KNN with k = 5 (instance-based, scale and outlier sensitive), and Decision Tree with max_depth = 10 (hierarchical, interpretable, scale-invariant). Distance-based SVM with RBF kernel represents algorithms critically dependent on distance calculations. Ensemble methods included Random Forest (100 trees), Gradient Boosting (100 estimators), XGBClassifier (100 rounds), and LightGBM (100 estimators), all representing scale-invariant, aggregation-based approaches.

3. Results with Analysis

Table 1 shows the results of classification based on the dataset on which no interventions were performed, or where no data preparation was carried out.

Table 1. Classification performance on the unprocessed dataset.

Classifier	Accuracy [%]	Precision [%]	Recall [%]	F1 Score [%]	AUC [%]
Random Forest	77.08	70.13	59.90	**63.93**	82.08
Decision Tree	70.95	58.42	58.00	58.14	67.82
XGBClassifier	73.04	62.52	57.12	59.12	78.51
Support Vector Machine	76.04	**74.33**	48.06	57.54	81.09
Logistic Regression	**77.33**	72.65	56.71	62.73	82.70
Gaussian Naive Bayes	75.77	66.94	60.09	63.00	81.11
KNN	71.22	60.34	51.87	55.01	74.41
Gradient Boosting	76.03	67.82	**60.13**	63.04	**83.53**
LightGBM	74.61	65.54	60.00	62.04	79.90

Given the missing values, varying feature value ranges, and outliers in the dataset, the relatively weak results across all metrics are as expected. The best results were achieved with Logistic Regression (accuracy: 77.33%, precision: 72.65%, recall: 56.71%, F1: 62.73%, and AUC: 82.70%), followed by Random Forest (accuracy: 77.08%, precision: 70.13%, recall: 59.90%, F1: 63.93%, and AUC: 82.08%) and Gradient Boosting (accuracy: 76.03%, precision: 67.82%, recall: 60.13%, F1: 63.04%, and AUC: 83.53%).

The next phase assessed the effect of imputing zeros using the mean and the median. Table 2 shows the results for mean and median zero-value imputation.

Table 2. Classification performance with zero-value imputation.

Classifier	Accuracy [%]	Precision [%]	Recall [%]	F1 Score [%]	AUC [%]
Imputation of zero values using the mean					
Random Forest	86.85	82.18	79.76	80.58	93.73
Decision Tree	83.98	76.25	78.65	77.10	82.78
XGBClassifier	87.49	82.97	80.71	81.48	94.60
Support Vector Machine	84.24	78.15	77.22	77.20	86.74
Logistic Regression	77.72	71.62	59.57	64.60	85.99
Gaussian Naive Bayes	77.20	67.99	64.12	65.61	85.22
KNN	84.11	79.42	72.35	75.10	88.78
Gradient Boosting	**88.41**	**84.99**	80.85	**82.55**	**95.07**
LightGBM	88.15	84.25	**81.26**	82.30	94.65
Imputation of zero values using the median					
Random Forest	87.62	83.52	80.79	81.79	93.81
Decision Tree	84.50	76.89	78.93	77.68	83.11
XGBClassifier	**87.89**	83.54	81.95	**82.42**	94.85
Support Vector Machine	85.01	76.29	**83.69**	79.44	88.61
Logistic Regression	77.46	71.68	58.66	63.70	84.83
Gaussian Naive Bayes	76.42	67.03	62.52	64.22	84.32
KNN	85.02	79.36	77.42	78.07	90.68
Gradient Boosting	87.76	**85.10**	79.92	81.90	**95.30**
LightGBM	87.76	83.62	81.66	82.26	94.84

Across most metrics, all classifiers improved substantially, except Logistic Regression and Gaussian Naive Bayes, which showed only marginal gains. Comparing mean and median zero-value imputation, median imputation generally yielded better or slightly better results for most classifiers, with the notable exceptions of Logistic Regression and Gaussian Naive Bayes. Overall, the best performing models were the ensemble models: Gradient Boosting, XGBClassifier, and LightGBM, closely followed by Random Forest, while the simpler models, Logistic Regression, and Gaussian Naive Bayes, performed worst.

The next phase involved analyzing model performance following zero imputation and scaling. Given that, per Table 2, median imputation performed better than mean imputation, Table 3 presents results with median imputation combined with standard and robust scaling.

Table 3. Classification performance with zero-value imputation and scaling.

Classifier	Accuracy [%]	Precision [%]	Recall [%]	F1 Score [%]	AUC [%]
StandardScaler					
Random Forest	87.49	83.48	80.43	81.59	94.40
Decision Tree	83.85	75.67	77.86	76.60	82.38
XGBClassifier	87.37	83.56	80.48	81.53	94.75
Support Vector Machine	84.37	78.99	75.49	76.77	90.77
Logistic Regression	76.94	70.44	57.80	62.76	85.28
Gaussian Naive Bayes	76.42	67.03	62.52	64.22	84.32
KNN	81.63	76.00	70.72	72.58	87.81
Gradient Boosting	**88.02**	**84.44**	81.39	**82.58**	**95.21**
LightGBM	87.76	83.38	**81.73**	82.21	94.73

Table 3. *Cont.*

Classifier	Accuracy [%]	Precision [%]	Recall [%]	F1 Score [%]	AUC [%]
			RobustScaler		
Random Forest	87.37	83.30	80.03	81.35	94.34
Decision Tree	83.98	76.16	77.53	76.71	82.41
XGBClassifier	87.37	83.56	80.48	81.53	94.75
Support Vector Machine	84.76	79.89	75.41	77.19	91.27
Logistic Regression	76.94	70.44	57.80	62.76	85.31
Gaussian Naive Bayes	76.42	67.03	62.52	64.22	84.32
KNN	83.46	76.03	76.11	75.71	89.11
Gradient Boosting	**88.15**	**84.77**	**81.39**	**82.72**	**95.22**
LightGBM	87.63	84.36	80.53	81.86	94.90

Overall, robust scaling yields better results, with the difference compared to standard scaling being particularly pronounced for algorithms such as KNN, which is highly sensitive to outliers. On the other hand, ensemble classifiers like Gradient Boosting, XGBClassifier, and Random Forest are inherently more robust to outliers and therefore perform well with standard scaling alone. The weakest performers, regardless of scaling type, remain Logistic Regression and Gaussian Naive Bayes.

After completing full data preparation, which included zero imputation and scaling, outlier handling using IQR and LOF methods was also performed in final phase. Table 4 shows the results of zero imputation with medians, robust scaling, and outlier handling.

Table 4. Classification performance with zero-value imputation, outlier handling, and scaling.

Classifier	Accuracy [%]	Precision [%]	Recall [%]	F1 Score [%]	AUC [%]
Random Forest	88.55	84.27	82.38	82.90	94.00
Decision Tree	82.50	73.83	77.78	75.38	81.47
XGBClassifier	89.08	84.99	**82.56**	83.52	94.83
Support Vector Machine	85.79	80.64	77.66	78.81	90.84
Logistic Regression	80.39	75.57	64.41	68.91	87.09
Gaussian Naive Bayes	81.05	72.79	71.81	71.85	87.76
KNN	84.47	76.84	79.25	77.64	89.73
Gradient Boosting	**89.21**	**86.55**	82.15	**83.91**	**94.99**
LightGBM	88.55	85.62	80.21	82.55	94.80

Overall, Gradient Boosting performed the best. XGBClassifier showed almost identical results, followed by somewhat weaker Random Forest and LightGBM. The weakest performers were again Logistic Regression and Gaussian Naive Bayes, which still showed noticeable improvement in this data preparation step, somewhat narrowing the gap to the other tested classifiers.

Since the performance of machine learning methods depends not only on data preparation and its outcomes but also on the manner in which models are trained, it is necessary to conduct hyperparameter tuning. This procedure involves repeatedly adjusting classifier parameters in various combinations with the aim of achieving optimal model performance and improved metric results. Hyperparameter tuning was carried out for all previously considered classifiers following the procedures described and implemented in final phase, with the results presented in Table 5.

Table 5. Classification performance with tuned hyperparameters.

Classifier	Accuracy [%]	Precision [%]	Recall [%]	F1 Score [%]	AUC [%]
Random Forest	89.08	84.52	**83.55**	83.67	94.47
Decision Tree	86.84	85.97	74.48	79.48	92.89
XGBClassifier	**89.21**	**86.10**	82.43	**83.89**	**95.58**
Support Vector Machine	85.79	79.96	78.74	78.97	90.68
Logistic Regression	80.26	74.25	65.87	69.22	87.26
Gaussian Naive Bayes	81.05	72.79	71.81	71.85	87.76
KNN	86.58	82.96	77.45	79.65	92.70
Gradient Boosting	88.29	84.50	81.34	82.53	95.25
LightGBM	88.68	84.45	82.24	82.95	95.51

The greatest benefit from hyperparameter tuning was observed in the Decision Tree, with accuracy improving by approximately 4%, precision by 12%, F1 by 4%, and AUC by 11%, while recall decreased by 3%. Overall, the best results were achieved by the XGBClassifier (accuracy: 89.21%, precision: 86.10%, recall: 82.43%, F1: 83.89%, AUC: 95.58%), followed by Random Forest (best recall: 83.55%, accuracy: 89.08%, precision: 84.52%, F1: 83.67%, AUC: 94.47%), LightGBM (accuracy: 88.68%, precision: 84.45%, recall: 82.24%, F1: 82.95%, AUC: 95.51%), and Gradient Boosting (accuracy: 88.29%, precision: 84.50%, recall: 81.34%, F1: 82.53%, AUC: 95.25%). However, most results with these models after hyperparameter tuning (except for AUC) are inferior to those prior to tuning. Figure 2 graphically depicts the aforementioned results.

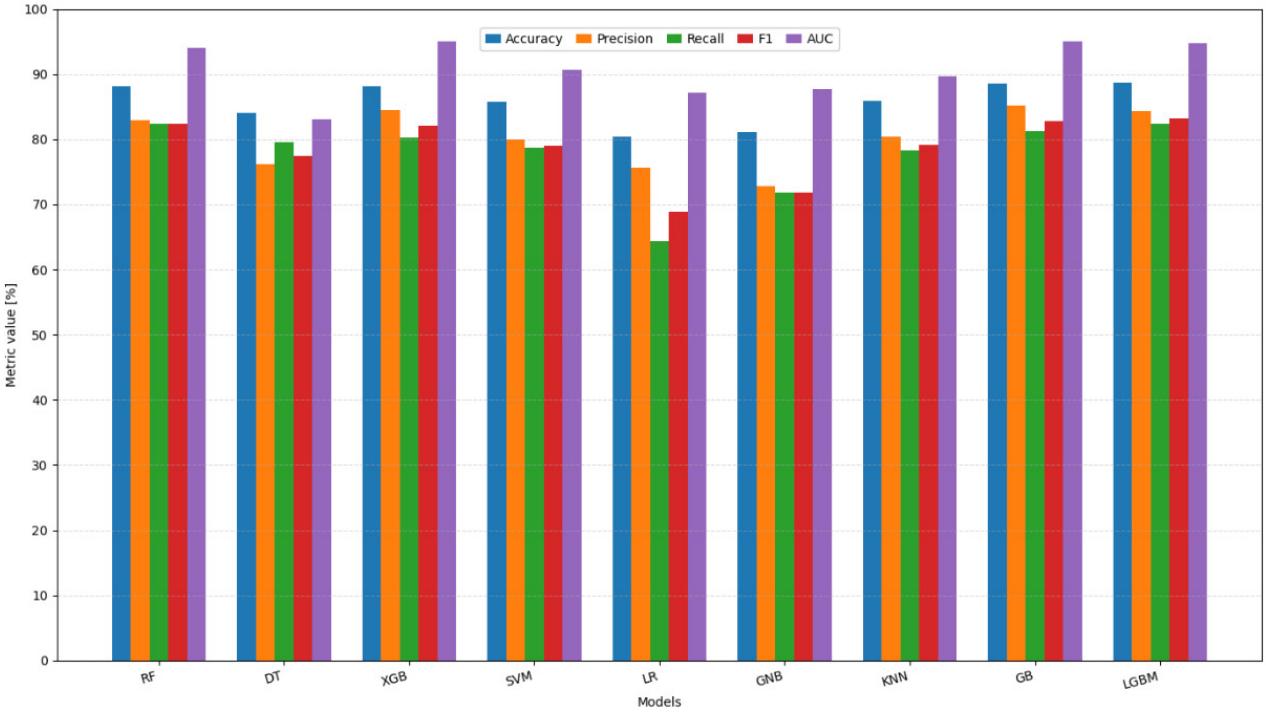


Figure 2. Model performance comparison (Accuracy, Precision, Recall, F1, AUC).

Given the class imbalance in the dataset, AUC was prioritized as the primary selection metric. From the figure, it is apparent that XGB, LightGBM, Gradient Boosting, and Random Forest achieve the highest AUC values.

https://doi.org/10.3390/engproc2026125013

4. Conclusions

The goal of the paper is to determine the effect of data preparation methods on various machine learning models for solving classification problems, as well as to compare classifiers with each other. The specific problem to which classification is applied in this study is predicting the risk of developing diabetes, one of the most common endocrinological diseases worldwide, which is continuously on the rise. Data preparation included methods such as imputing missing values, removing outliers, and data scaling.

The results showed that the most significant improvement across all metrics and for almost all classifiers, except Logistic Regression and Gaussian Naive Bayes, was recorded through imputation of missing values, primarily using medians. Other methods, namely outlier removal and data scaling, only slightly improved model performance, with the exception that Logistic Regression and Gaussian Naive Bayes showed notable improvement only after outlier removal. Regarding scaling methods, RobustScaler generally gave better results, especially for simpler classifiers more sensitive to outliers, while for the more complex ensemble classifiers, StandardScaler was almost equally effective. Generally, ensemble models such as Gradient Boosting, XGBClassifier, LightGBM, and Random Forest showed better results, and due to their inherent properties, no significant differences were observed between zero imputation with medians versus means, or between robust versus standard scaling. On the other hand, simpler models like Logistic Regression, KNN, and SVM showed greater improvements with zero imputation by medians and robust scaling compared to mean imputation and standard scaling, indicating that simpler models are more sensitive to unprepared data and therefore require higher-quality data preparation. Advanced ensemble models are less sensitive and achieve similar results with simpler preparation methods. This confirms the importance of high-quality and comprehensive data preparation as a prerequisite for building effective models. The choice of classifier depends on the data preparation; it is evident that simpler classifiers require more complex preparation methods, while more complex classifiers achieve comparable results with robust and simpler data preparation methods.

This study was conducted on only a single dataset, which represents a limitation to some extent, but at the same time opens up opportunities for further research on the effect of the applied preprocessing methods on other types of datasets. Additionally, the work employed only foundational baselines. Beyond these, some advanced methods could also be considered, including Extra Trees, regularized linear models (ElasticNet, Lasso, Ridge), and Linear Discriminant Analysis (LDA).

Author Contributions: Conceptualization, G.M. and I.I.; methodology, G.M. and D.V.; software, I.I. and D.V.; validation, G.M. and D.V.; formal analysis, I.I.; investigation, I.I. and D.V.; resources, T.B.; data curation, T.B.; writing–original draft preparation, G.M., I.I., D.V. and T.B.; writing–review and editing, G.M., I.I., D.V. and T.B.; visualization, I.I.; supervision, G.M. and D.V.; project administration, G.M.; funding acquisition, I.I. Author G.M. passed away prior to the publication of this manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union-NextGenerationEU.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: This research was performed using the Advanced Computing service provided by University of Zagreb University Computing Centre-SRCE.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Amilo, D.; Sadri, K.; Hincal, E.; Farman, M.; Nisar, K.S.; Hafez, M. An Integrated Machine Learning and Fractional Calculus Approach to Predicting Diabetes Risk in Women. *Healthc. Anal.* **2025**, *8*, 100402. [CrossRef]
2. Halperin, I.J.; Wicklow, B.; Amed, S.; Chambers, A.; Courage, C.; Cummings, E.; Kirkland, P.; MacKay, D.; Nakhla, M.; Punthakee, Z.; et al. Glycemic Management Across the Lifespan for People with Type 1 Diabetes: A Clinical Practice Guideline. *Can. J. Diabetes* **2025**, *49*, 5–18. [PubMed]
3. Albanese, I.; Rendon, L.; Nakhla, M.; Rahme, E.; Dasgupta, K. Gestational Diabetes and Parental Type 2 Diabetes as Risk Indicators for Type 1 Diabetes in Offspring: A Systematic Review and Meta-Analysis. *eClinicalMedicine* **2025**, *85*, 103321. [CrossRef] [PubMed]
4. Jiang, L.; Xia, Z.; Zhu, R.; Gong, H.; Wang, J.; Li, J.; Wang, L. Diabetes Risk Prediction Model Based on Community Follow-up Data Using Machine Learning. *Prev. Med. Rep.* **2023**, *35*, 102358. [CrossRef]
5. Munagala, V.; Thudumu, S.; Logothetis, I.; Bhandari, S.; Vasa, R.; Mouzakis, K. A Comprehensive Survey on Machine Learning Applications for Drilling and Blasting in Surface Mining. *Mach. Learn. Appl.* **2024**, *15*, 100517. [CrossRef]
6. Pima Indians Diabetes Database. Available online: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database (accessed on 11 March 2025).
7. Alam, S.; Ayub, M.S.; Arora, S.; Khan, M.A. An Investigation of the Imputation Techniques for Missing Values in Ordinal Data Enhancing Clustering and Classification Analysis Validity. *Decis. Anal. J.* **2023**, *9*, 100341. [CrossRef]
8. Unmasking the Outliers: Exploring the Interquartile Range Method for Reliable Data Analysis. Available online: https://procogia.com/interquartile-range-method-for-reliable-data-analysis/ (accessed on 30 October 2025).
9. Breunig, M.; Kröger, P.; Ng, R.; Sander, J. LOF: Identifying Density-Based Local Outliers. *ACM Sigmod Record* **2000**, *29*, 93–104. [CrossRef]
10. Rashmi, C.R.; Shantala, C.P. Evaluating Deep Learning with Different Feature Scaling Techniques for EEG-Based Music Entrainment Brain Computer Interface. *e-Prime-Adv. Electr. Eng. Electron. Energy* **2024**, *7*, 100448. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.