ENHANCING EARLY-STAGE DIABETES PREDICTION USING DATA MINING ALGORITHMS AND NORMALISATION TECHNIQUES

Ravikant Kholwal
PDPM IIITDMJ, Jabalpur, India

ABSTRACT

The global rise in diabetes prevalence is a pressing health concern, emphasizing the critical need for timely interventions and effective diagnostic strategies. Early detection of diabetes is of paramount importance as it allows for timely interventions, potentially preventing or delaying the onset of complications associated with the disease. Early-stage diabetes detection can lead to better management and control of the condition, reducing the risk of severe complications and improving the overall quality of life for patients. In the realm of medical research, data mining algorithms have emerged as powerful tools for extracting meaningful patterns and insights from vast amounts of data. These algorithms, when applied to medical datasets, have the potential to revolutionize the way diseases like diabetes are detected and managed. The focus of this study is to explore the potential of detecting early-stage diabetes using attributes that are not strictly medical in nature. Such an approach can broaden the scope of diabetes detection, making it more accessible and potentially more efficient. To achieve this, the study delves into the evaluation of various data normalization techniques. Normalization is a crucial step in data preprocessing, ensuring that the features in a dataset are on a comparable scale. This is vital for the performance of many machine learning algorithms, as features on different scales can unduly influence the outcome. Several algorithms were employed in this study, including Naive Bayes, K-Nearest Neighbour (KNN), Support Vector Machines (SVM), Decision Tree, Random Forest, and Gradient Boosting Classifier (GBC). These algorithms were applied to the Early Stage Diabetes Risk Prediction Dataset, which was preprocessed using normalization methods like Decimal Point Scaling, Z-Score Normalisation, and Pareto Scaling, to name a few. The results of this study are promising. Notably, even without relying on traditional medical diagnostic data, early-stage diabetes prediction was achievable. The Gradient Boosting Classifier (GBC), when combined with the right data normalization technique, stood out among the algorithms, achieving a prediction accuracy rate of 99.038%. This accuracy is a testament to the potential of data-driven approaches in medical research and their ability to provide valuable insights into disease detection and management.

KEYWORDS: *Diabetes, Decision Tree, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbour (KNN), Gradient Boosting Classifiers (GBC), Machine Learning*

I. INTRODUCTION

Data mining refers to the process of extracting valuable information and uncovering hidden patterns from large datasets. Its primary goal is computationally extracting meaningful insights and presenting them in an understandable format; while data mining's secondary aim is allowing prediction, description, and an interpretive representation of future values from large sets.

At the forefront of disease diagnosis and treatment is extracting relevant data from large datasets in order to facilitate efficient decision-making processes and facilitate optimal decision-making processes. Data mining in medicine seeks to create automated tools which detect diseases while informing both patients and physicians of disease severity as well as treatment options based on patient history and symptoms.

DOI: 10.5281/zenodo.10141933

Data mining techniques offer valuable insights into medicine, helping discover hidden patterns among medical datasets. Data mining techniques also facilitate disease prediction using these same techniques; that is, using them to analyze and predict whether an individual patient may become susceptible to specific illnesses; this allows doctors to make more informed decisions by decreasing testing requirements while inferring hierarchies among various diseases and tests.

Studies have been done using data mining techniques to detect various diseases, such as heart disease, kidney disease, hepatitis, lung cancer, liver disorders, breast cancer and thyroid illness. Diabetes, being one of the more prevalent illnesses, can greatly benefit from early detection and subsequent prevention measures; by analyzing symptoms, one may show the likelihood of being affected by diabetes can be determined.

Hospitals collect vast amounts of patient data that are often underutilized to improve patient treatment. By employing data mining, hospitals can use this collected information to detect diseases like diabetes early. As part of its preprocessing steps, data normalization reduces redundancies and increases integrity - as well as irregularities related to data exploration, such as deletion, insertion and updates.

This analysis aims to demonstrate whether diabetes can be identified in its initial stages without medical analysis reports. We employed data mining techniques along with various normalization techniques - the goal being evaluating their influence on classifier performance.

II. RELATED WORK

Kandhasamy and Balamurali [1] conducted experiments comparing various machine learning (ML) techniques used for diabetes prediction with the aim of producing accurate and reliable results. J48 achieved the highest accuracy at 73.82% without preprocessing, while KNN (k = 1) and Random Forest achieved 100% after preprocessing.

Yuvaraj and Sripreethaa [2] utilized Random Forest, Decision Tree and Naive Bayes algorithms for diabetes prediction using the preprocessed Pima Indian Diabetes dataset. Their Random Forest algorithm achieved the highest accuracy rate with a 94% accuracy rate.

Tafa et al. [3] developed an improved integrated model consisting of SVM and Naive Bayes for diabetes prediction on data collected in three locations throughout Kosovo. Their combined algorithms achieved an accuracy rate of 976.6% - surpassing their individual performances of SVM (95.52%) and Naive Bayes (94.52%).

Deepti and Dilip [4] applied Decision Tree, SVM and Naive Bayes classifiers on the Pima Indian dataset, with Naive Bayes reaching maximum accuracy at 76.30%.

Mercaldo et al. [5] applied J48, Multilayer Perceptron, HoeffdingTree, JRip, BayesNet and RandomForest classifiers on the Pima Indian dataset with Hoeffding Tree algorithm showing superior precision-recall F measure values at 0.757 for precision, recall and F measure values respectively.

Negi and Jaiswal [6] investigated the application of SVM on a combined dataset consisting of Pima Indians and Diabetes 130-US patients. After preprocessing, their SVM model reached an accuracy rate of 72%.

Olaniyi and Adnan [7] successfully employed a Multilayer Feed-Forward Neural Network with a backpropagation algorithm on the Pima Indian Diabetes Database to achieve an accuracy rate of 82%.

Soltani and Jafarian [8] employed the Probabilistic Neural Network (PNN) with an accuracy rate of 89.56% in training and 81.49% for testing purposes on the Pima Indian dataset, respectively.

Rakshit et al. [9] utilised a Two-Class Neural Network on a Pima Indian dataset preprocessed before running their classification algorithms and achieved an accuracy rate of 83.3 percent.

Mamuda and Sathasivam [10] evaluated three supervised learning algorithms on the Pima Indian dataset: Levenberg Marquardt (LM), Bayesian Regulation (BR) and Scaled Conjugate Gradient

(SCG). Of these algorithms, Levenberg Marquardt (LM) demonstrated the highest performance with an MSE value of only 0.00025091.

Equivalence Class Clustering and bottom-up Lattice Traversal algorithm, talked about in [11], can be employed for determining both controlled and uncontrolled diabetes states. Eclat, being a depth-first-based algorithm, efficiently generates patterns on compact datasets.

[12] Furnishes an evaluative comparison of various standardization methods, such as Min-Max Normalization, Z-Score Normalization, and Decimal Scaling Normalization, utilizing K-Means Clustering Algorithm as the benchmark. Findings indicate that Min-max normalization surpassed Z-Score and Decimal Scaling Normalization methods in terms of misclassification errors.

[13] Showcases an analysis delving into data normalization methods and their effect on intrusion classifier efficacy, particularly focusing on Rational Normalization, Mean Range Normalization, Maximize Normalization, Frequency Normalization, and Hybrid Normalization techniques. The detailed examination of these normalization techniques underscores their profound influence on performance metrics like intrusion classification rate and processing duration.

[14] Explores the role of data mining in healthcare. This research involves surveying journals and publications from disciplines including medicine, health, computer science and engineering - this showcases its applicability within health sector applications while emphasising its significance.

Amid the mounting global diabetes crisis, there's an emergent need for innovative diagnostic tools. While the importance of early diabetes detection is undisputed, relying solely on conventional medical data might be limiting. Recent advancements in data science suggest that non-medical datasets might hold the key to early diagnosis. Yet, the influence of data normalization on the efficacy of these predictive algorithms is still a gray area. This research endeavors to bridge this gap by evaluating the synergy between diverse classification algorithms and normalization methods, using the "Early-stage diabetes risk prediction dataset" from UCI as a testbed.

III. METHODOLOGY

Predicting diseases in their early stages is of paramount importance in the medical field. Early-stage prediction not only allows for timely interventions but also increases the chances of successful treatment outcomes. In the context of diabetes, early detection can lead to better management and control of the condition, reducing the risk of severe complications and improving the overall quality of life for patients.

In this research endeavour, the focus is on utilizing the "Early-stage diabetes risk prediction dataset" sourced from the UCI repository. This dataset is rich with indicators, including various signs and symptoms, that can provide insights into whether an individual might be in the initial phases of diabetes. The primary objective is to discern which machine learning classification algorithms can most effectively and accurately predict the onset of early-stage diabetes based on this dataset.

A significant portion of the study is also dedicated to understanding the influence of different normalization techniques on the accuracy of classification. Normalization is a pivotal step in data preprocessing, ensuring that the features in a dataset are on a comparable scale, which is vital for the optimal performance of many machine learning algorithms. The overarching goal is to pinpoint the most synergistic pairing of a classification algorithm and a normalization method that together maximize the prediction accuracy for early-stage diabetes. This entire process is visually represented in Figure 1, which provides a schematic overview of the steps involved in early-stage diabetes prediction.

Data preprocessing is an essential phase in any data-driven study. In this context, the dataset's categorical attributes are transformed into numerical values using binary encoding. This transformation ensures that machine learning algorithms, which typically require numerical input, can process the data. Subsequent to this preprocessing step, a gamut of data mining algorithms, including but not limited to naive Bayes, KNN with adjustable neighbour configurations, support vector machine, decision tree, random forest, and gradient boosting classifiers, are employed. These algorithms are tested on the dataset, both with and without the application of various normalization techniques. The outcomes of these experiments are then meticulously analysed to determine the most potent combination of a classifier and normalization method that delivers superior prediction results.

IV. IMPLEMENTATION

Implementation in this study includes working with an early-stage diabetic risk prediction dataset consisting of 520 instances and 17 attributes, such as class level. A more in-depth description can be found in Table 1.

To aid data analysis, all attributes other than age are represented as either yes or no values. As part of preprocessing steps, categorical attributes are converted to numerical values between zero (0) and one (1) for analysis purposes.

Figure 2 depicts a class-level distribution of this dataset that provides insight into proportions within each class.

As part of a normalization technique, decimal point scaling is a widely utilized strategy. This technique normalizes each attribute by dividing its value against its respective maximum among all attribute values, effectively shifting their decimal points. Decimal point scaling can be especially useful when dealing with datasets that experience logarithmic fluctuations in feature values to ensure the comparability of attributes. Each value $X_{i,n}$ of the given data is rescaled into $X'_{i,n}$ as shown in equation (1):

$$X'_{i,n} = \frac{X_{i,n}}{10^{j}} \qquad (1)$$

Where, J = $\log_{10}$ (max (X_i)) and max (X_i) is the maximum value of the series.

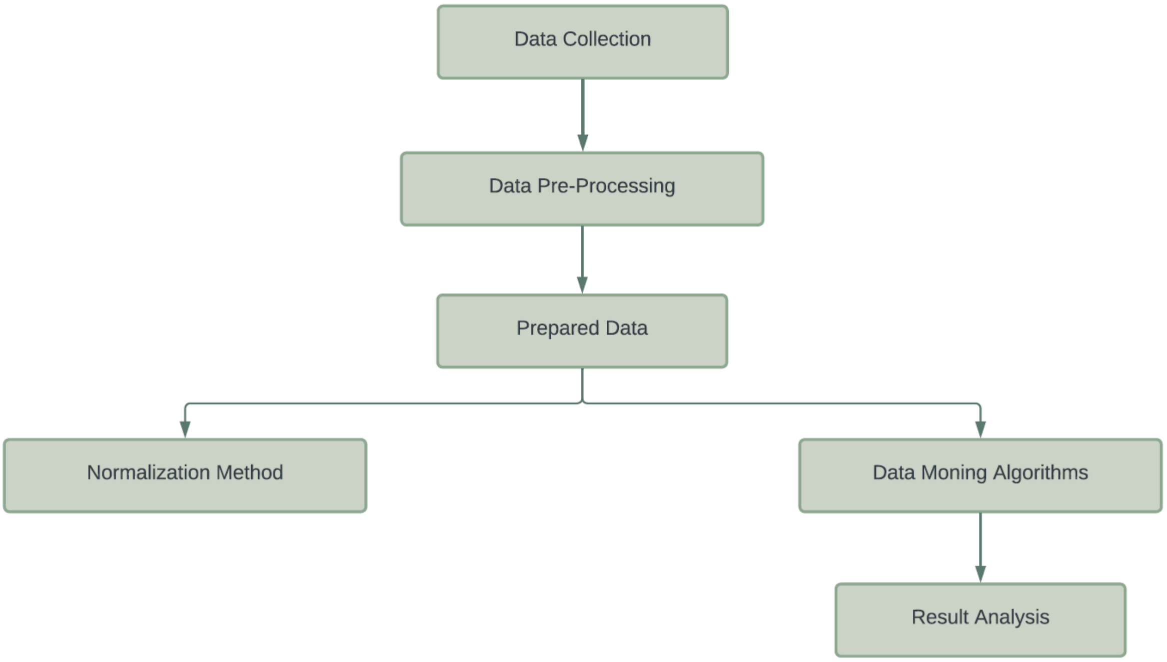


Figure 1 Experimental Methodology

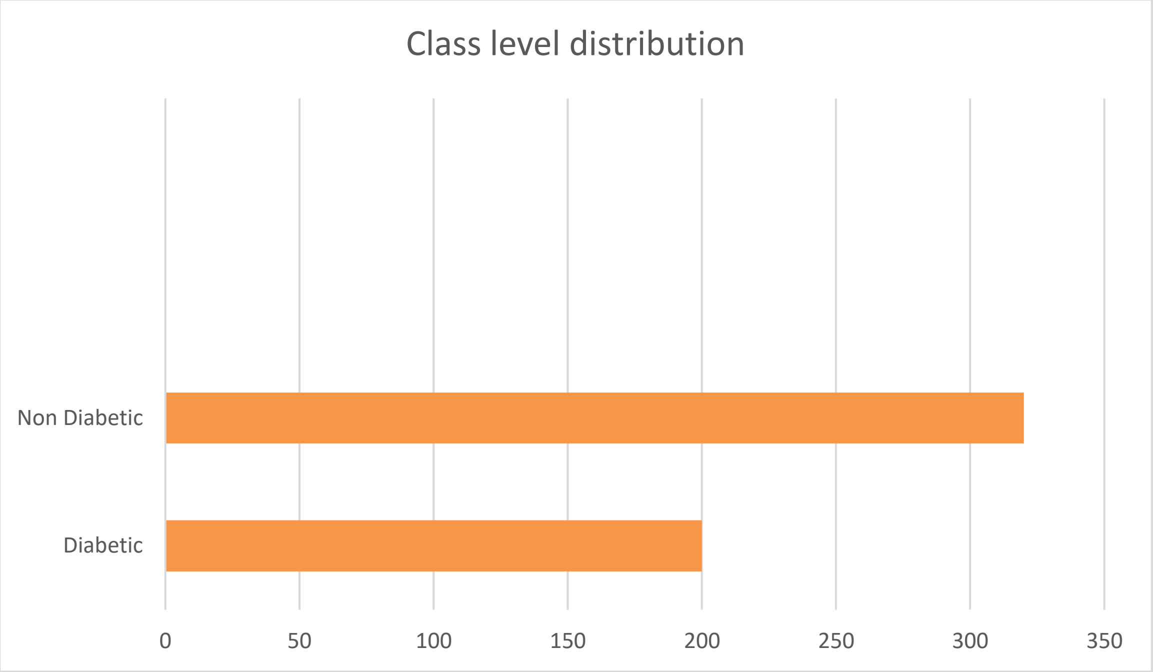


Figure 2 Early-stage diabetes dataset class level distribution

Table 1. Attribute description of Early – Stage Diabetes Risk Prediction Dataset

Attribute Name	Attribute Type	Attribute Details
Age	Numerical	Age (years)
Gender	Binary	Gender
Polyuria	Binary	Large volumes of urine pass
Polydipsia	Binary	Excessive thirst or excess drinking
Sudden weight loss	Binary	Suddenly lost weight
Weakness	Binary	Physically feels weak
Polyphagia	Binary	Excessive hunger
Genital thrush	Binary	If the patient has genital thrush
Visual blurring	Binary	If the patient has blurry eyesight
Itching	Binary	Itchy skin
Irritability	Binary	If the patient gets irritated easily
Delayed healing	Binary	Delayed healing of wounds
Partial paresis	Binary	If the patient has weak muscle
Muscle stiffness	Binary	Difficult to move muscle
Alopecia	Binary	Hairs falls in small patches
Obesity	Binary	An excessive amount of body fat
Class	Class level	If a patient is diabetic or not

Z-Score Normalization, often termed as standard score normalization, is a widely-accepted data preprocessing technique. At its core, this method aims to adjust the scale of features in a dataset, ensuring that they have a mean (average) value of zero and a standard deviation of one. The primary motivation behind this transformation is to bring diverse features, which might have different scales or units, onto a common scale. This is particularly crucial for algorithms that are sensitive to feature scales, ensuring that no particular feature unduly influences the model due to its scale.

The mathematical transformation involved in Z-Score Normalization is relatively straightforward. For a given data point, $X_{i,n}$, in the dataset, the transformed or normalized value, $X'_{i,n}$ is computed using:

$$\mathrm{X'}_{\mathrm{i,n}} = \frac{\mathrm{X_{i,n}} - \mu_{\mathrm{i}}}{\sigma_{\mathrm{i}}} \qquad (2)$$

In this equation, μ_i represents the mean of the feature, and σ_i denotes its standard deviation. The subtraction of the mean ensures that the transformed feature has a mean of zero, while the division by the standard deviation scales the feature such that its standard deviation becomes one.

Pareto scaling is another technique that shares similarities with Z-Score Normalization. The key difference lies in the scaling factor: while Z-Score uses the standard deviation, Pareto scaling employs its square root . This subtle difference can lead to varied results, especially when dealing with features that have outliers or non-normal distributions.

$$X'_i = \frac{X_i - \mu_i}{\sqrt{\sigma_i}} \qquad (3)$$

The choice of normalization or scaling technique can have profound implications on the performance of machine learning or data mining algorithms. For instance, Mean and standard deviation scaling, represented by μ_i and σ_i, can enhance the prominence of features that might otherwise be overshadowed due to their lower concentrations. This method offers a balance – it amplifies certain features while also mitigating noise, leading to more robust models.

Building upon the foundational concepts of Z-Score Normalization, Variable Stability Scaling introduces an additional layer of complexity by incorporating the Coefficient of Variation (CV) as a scaling factor. This metric, CV, provides insights into the relative variability of a feature. By considering this variability, the normalization process becomes more adaptive and can cater to features with diverse distributions and scales. The coefficient of variation is essentially a normalized measure of dispersion, defined as the ratio of the standard deviation to the mean. This ratio offers insights into the relative variability of a dataset, making it a valuable tool in the normalization process.

$$X'_{i,n} = \frac{X_{i,n} - \mu_i}{\sigma_i} \times (\mu_i - \sigma_i) \qquad (4)$$

For optimal data preprocessing, the coefficient of variation (CV) should be used to assign different levels of importance to different features. It places greater weight on features with smaller standard deviations while diminishing their significance with larger standard deviations, providing a more balanced representation of data.

Min-max normalization is a fundamental data preprocessing technique that aims to adjust the scale of features in a dataset. The primary objective of this method is to transform features so that they fall within a specified range, typically between 0 and 1. This is achieved by taking into account the minimum and maximum values of the original feature.

Mathematically, the transformation is represented as:

$$X'_{i,n} = \frac{X_{i,n} - min(X_i)}{max(X_i) - min(X_i)} \times (nMax - nMin) + nMin \qquad (5)$$

Where:

- $X_{i,n}$ is an original data point.
- $X'_{i,n}$ is the normalized data point.
- $min(X_i)$ represents the minimum value of the feature.
- $max(X_i)$ represents the maximum value of the feature.
- $nMax$ and $nMin$ are the desired lower and upper bounds for the rescaled data.

Max normalization is an extension of the min-max normalization technique. Instead of rescaling features to lie between 0 and 1, max normalization adjusts them to fall within the interval of -1 to 1. This is particularly useful in contexts where data symmetry around zero is desired, such as in certain neural network applications.

The transformation for max normalization is given by:

$$X'_{i,n} = \frac{X_{i,n}}{Max(|X_i|)} \tag{6}$$

Here:

- $\mathrm{X'}_{\mathrm{i,n}}$ is the normalized data point.
- $Max(|X_i|)$ denotes the maximum absolute value of the feature.

Both min-max and max normalization techniques are widely used in machine learning and data analytics to ensure that features are on a consistent scale, which is crucial for algorithms that are sensitive to feature magnitudes. By ensuring that all features are on a similar scale, these normalization techniques help in improving the convergence speed of learning algorithms and can lead to better performance in many scenarios.

Normalization or standardization of data is a pivotal step in many data processing pipelines, especially in contexts where the scale or distribution of features can significantly impact the performance of algorithms. One of the primary goals of normalization is to adjust the scale of features so that they are comparable and do not unduly influence the outcome due to their original scale or distribution.

In the realm of normalization techniques, methods based on the standard deviation and mean of data are quite prevalent. The rationale behind these techniques is rooted in statistics. By adjusting data based on its mean and standard deviation, one can ensure that the transformed data has a mean of zero and a standard deviation of one. This kind of transformation is particularly useful when the relationships within the data are of interest, as it preserves these relationships even as the underlying statistical properties of the data evolve over time. However, a notable challenge with these methods is their sensitivity to extreme values or outliers. Since both mean and standard deviation are influenced by outliers, normalization based on these metrics can sometimes lead to skewed or misleading results if the original data contains extreme values.

Another approach to normalization is the Maximum Absolute Scaling method. This method is relatively straightforward and is based on scaling data according to its maximum absolute value. The primary advantage of this method is its simplicity and its ability to scale data between -1 and 1, making it suitable for algorithms that expect input features in this range.

Mathematically, the Maximum Absolute Scaling method can be represented as:

$$X' = \frac{X}{Max(X)} \tag{7}$$

Here:

- X' is the normalized value.
- X is the original data point.
- Max(X) represents the maximum absolute value of the dataset.

By dividing each data point by the maximum absolute value, this method ensures that the transformed data lies between -1 and 1. This range is particularly useful for certain algorithms and ensures that no single feature dominates the outcome due to its scale.

In summary, while normalization techniques based on mean and standard deviation offer the advantage of preserving relationships within data, they can be sensitive to outliers. On the other hand, methods like Maximum Absolute Scaling provide a straightforward way to scale data , making it suitable for a range of applications. The choice of normalization method often depends on the specific requirements of the task at hand and the nature of the data being processed.

As a result of performing the previous conversion, values can range between -1 to 1. To further process this data, Mean Centered Scaling is employed. This technique removes any offset in the data by subtracting each instance of an element from its mean, centring around zero, allowing for enhanced analysis and interpretation.

Mean subtraction is a fundamental preprocessing step in data analysis, especially in contexts where centering the data around zero is crucial. By subtracting the mean of a dataset from each data point, the transformed data will have a mean of zero. This is particularly beneficial for algorithms that are sensitive to data not centered around zero, as it ensures that the average value of the features is zero.

Mathematically, the mean subtraction method can be represented as:

$$X'_{i,n} = \quad X_{i,n} - \mu_i \qquad (8)$$

Where:

- $X_{i,n}$ is the original data point for the attribute.
- $X'_{i,n}$ is the mean-centered data point.
- μ_i denotes the mean of the dataset for the attribute.

By centring the data around zero, this method helps in improving the convergence speed of learning algorithms and can lead to better performance in many scenarios. Ifs a foundational step in many advanced normalization techniques and serves as a precursor to other transformations, such as dividing by the standard deviation to achieve standard score normalization.

Softmax Normalization is a nonlinear normalization technique designed to address cases in which data distribution is uneven and linear scaling cannot be applied. Logarithmic, sigmoid and exponential functions may be utilized instead to map data into specific intervals; typically ranging from 0-1 values. Equation (9) depicts this process where each value x is transformed into its equivalent "x'", and the sum is used as the indicator term

$$X' = \frac{e^x}{sum\ (e^x)} \qquad (9)$$

The Power Transformation technique is used to address heteroscedasticity in data. Its goal is to transform heteroscedasticity-laden statistics into homoscedastic form by calculating the square root of data before using mean-centred rescaling and then recalculating square roots again, as shown by Equation (10) which illustrates this transformation process where each attribute, $X_{i,n}$ of the data is converted into $X'_{i,n}$. $P_{i,n} = \sqrt{\hat{X}_{i,n}}$ and μ_i^p denotes the mean value of $P_{i,n}$

$$X'_{i,n} = P_{i,n} - \mu_i^p \qquad (10)$$

Power transformation is a mathematical approach aimed at stabilizing variance, making the data more closely follow a Gaussian distribution, and making the data more interpretable. However, a limitation of this transformation is its inability to handle negative values. To circumvent this, a shift operation is often applied to each data value, Xi, n, before normalization. This shift ensures that all values are positive by subtracting the minimum value of the series from each data point, as represented in equation (11):

$$\widehat{X_{i,n}} = \quad X_{i,n} - Min(X_i) \qquad (11)$$

The Mean Absolute Deviation (MAD) Normalization method is a robust technique that focuses on rescaling data based on its median and the Mean Absolute Deviation values for each feature. The primary advantage of using MAD over standard deviation is its resilience to outliers. By centering the data around its median and scaling based on MAD, this normalization method ensures that the transformed data is less influenced by extreme values. The normalization process using MAD is given by equation (12):

$$X'_{i,n} = \frac{X_{i,n} - med_i}{MAD_i} \tag{12}$$

Where med_i represents the median value of the feature and MAD_i is the Mean Absolute Deviation.

Robust Scaling is another normalization technique that emphasizes dealing with outliers effectively. By considering both the mean and the Median Absolute Deviation (MAD), Robust Scaling aims to standardize data such that its mean is zero and its dispersion is standardized. This method is particularly effective when the data follows a Gaussian distribution. The transformation using Robust Scaling is represented in equation (13):

$$X'_{i,n} = \frac{X_{i,n} - Q_1(X)}{Q_3(X) - Q_1(X)} \tag{13}$$

Where Q_1 and Q_3 represent the 1st and 3rd quartiles, respectively.

Log Scaling is another transformation technique that focuses on compressing the scale of data, especially when there's a wide range of values. By taking the logarithm of each value, the distribution of data points can be made more uniform, which can be beneficial for linear models. The transformation using Log Scaling is given by equation (14):

$$\mathrm{X}' = \log(\mathrm{X}) \tag{14}$$

In the context of this study, Python3 was the chosen programming language, complemented by the Scikit-learn machine learning library. Jupyter Notebooks served as the primary integrated development environment (IDE). Six distinct classification algorithms were employed on the dataset, including Naive Bayes, K-nearest Neighbours (KNN) with variable neighbour settings, Support Vector Machine (SVM), Decision Tree, Random Forest Classifier, and Gradient Boosting Classifier. Notably, the Naive Bayes classifier operates on the principle of Bayes' theorem and assumes attribute independence, making it a probabilistic classifier.

By integrating these techniques and tools, the study aimed to provide a comprehensive analysis of the dataset and derive meaningful insights.

K-nearest neighbours (KNN) is an algorithm designed to predict class labels based on majority votes from near neighbours, wherein k represents the number of neighbours being considered for classification purposes.

Support Vector Machine (SVM) is a kernel-based algorithm used to distinguish among various labelled data points by creating a decision boundary.

The Decision Tree algorithm constructs a tree-like structure by evaluating entropy and information gain, classifying different labelled data points into groups. It comprises root nodes, internal nodes, and leaf nodes, where each leaf node represents an ultimate classification result.

Random Forest is an ensemble method that combines multiple decision trees in order to make predictions based on the majority vote.

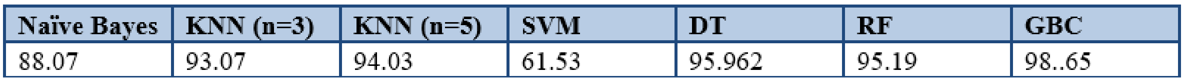
Gradient Boosting Classifier is a classification technique that seeks to minimize error from previous iterations by gradually adding weak learners using gradient descent, typically decision trees, as weak learners in this approach

V. RESULTS

The classification accuracy of each classifier was measured both before and after applying normalisation techniques. Accuracy measurement was conducted using the ten-fold cross-validation technique. The results obtained are presented in Table 2, which showcases the accuracy of early-stage diabetes risk prediction without applying any normalisation techniques. Additionally, Table 3 displays the accuracy of early-stage diabetes risk prediction after applying various normalisation techniques.

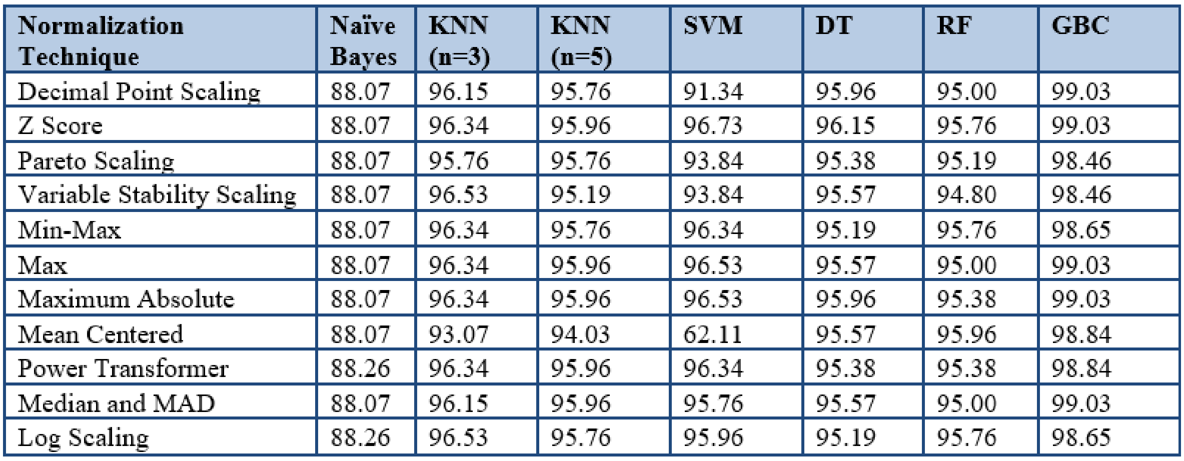
Tables 2 and 3 provide evidence that data mining algorithms are capable of effectively detecting diabetes at an early stage, though their level of accuracy varies between algorithms. GBC achieved the highest prediction accuracy with 95.962% accuracy, outperforming other algorithms by 2.7%. The Random Forest classifier achieved an accuracy rate of 95.192% and, when combined with data normalisation techniques, even greater precision.

Table 2. Early – stage diabetes risk prediction accuracy without normalization



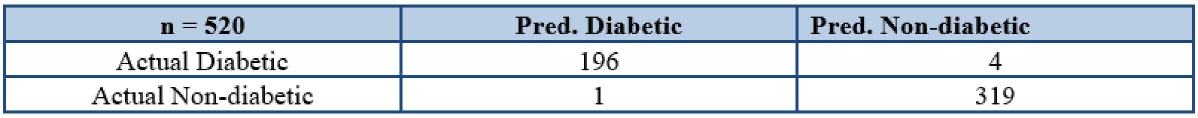
Naïve Bayes	KNN (n=3)	KNN (n=5)	SVM	DT	RF	GBC
88.07	93.07	94.03	61.53	95.962	95.19	98..65

Table 3. Prediction accuracy for early-stage diabetes risk prediction dataset with normalization



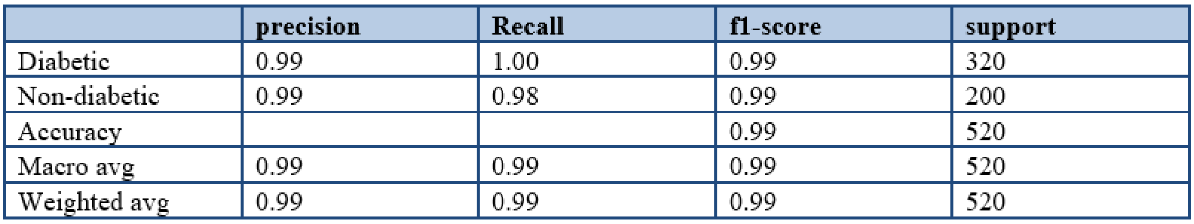
Normalization Technique	Naïve Bayes	KNN (n=3)	KNN (n=5)	SVM	DT	RF	GBC
Decimal Point Scaling	88.07	96.15	95.76	91.34	95.96	95.00	99.03
Z Score	88.07	96.34	95.96	96.73	96.15	95.76	99.03
Pareto Scaling	88.07	95.76	95.76	93.84	95.38	95.19	98.46
Variable Stability Scaling	88.07	96.53	95.19	93.84	95.57	94.80	98.46
Min-Max	88.07	96.34	95.76	96.34	95.19	95.76	98.65
Max	88.07	96.34	95.96	96.53	95.57	95.00	99.03
Maximum Absolute	88.07	96.34	95.96	96.53	95.96	95.38	99.03
Mean Centered	88.07	93.07	94.03	62.11	95.57	95.96	98.84
Power Transformer	88.26	96.34	95.96	96.34	95.38	95.38	98.84
Median and MAD	88.07	96.15	95.96	95.76	95.57	95.00	99.03
Log Scaling	88.26	96.53	95.76	95.96	95.19	95.76	98.65

Table 4. Confusion Matrix for GBC Classifier



n = 520	Pred. Diabetic	Pred. Non-diabetic
Actual Diabetic	196	4
Actual Non-diabetic	1	319

Table 5. Precision, recall and f1-score for GBC classifier



	precision	Recall	f1-score	support
Diabetic	0.99	1.00	0.99	320
Non-diabetic	0.99	0.98	0.99	200
Accuracy			0.99	520
Macro avg	0.99	0.99	0.99	520
Weighted avg	0.99	0.99	0.99	520

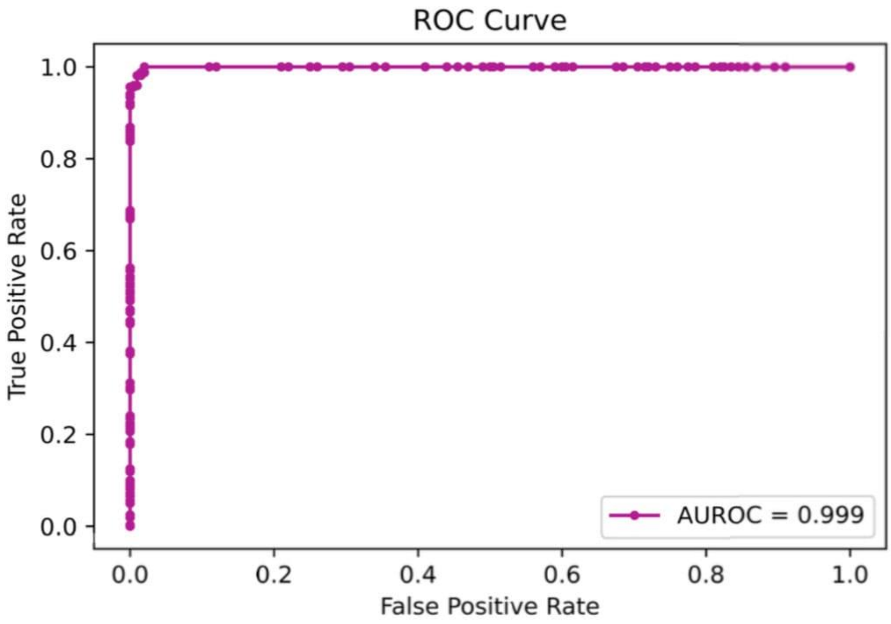


Figure 3 - ROC curve for GBC classifier

Notably, the GBC classifier exhibited even greater precision. Decimal point scaling, Z-score normalization, max normalization, maximum absolute scaling, and median and Mean Absolute Deviation (MAD) normalization are data preprocessing techniques that aim to adjust the scale of features in a dataset. These normalization methods are designed to transform features so that they are comparable and do not unduly influence the outcome due to their original scale or distribution. In the context of this study, these normalization techniques, when applied, resulted in a remarkable prediction accuracy of 99.038%. This accuracy level is notably higher, with an improvement of 0.4%, compared to the Gradient Boosting Classifier (GBC) without normalization.

The Gradient Boosting Classifier (GBC) is a machine learning algorithm that builds an additive model in a forward stage-wise fashion. It generalizes the boosting method by allowing optimization of an arbitrary differentiable loss function. In this study, the GBC classifier's performance was evaluated using various metrics, and its results were documented in tables for clarity and ease of interpretation.

Table 4 provides a detailed view of the GBC classifier's performance using a confusion matrix. A confusion matrix is a table layout that visualizes the performance of an algorithm, typically a supervised learning one. It's a vital tool in machine learning to understand the true positives, true negatives, false positives, and false negatives produced by the classifier.

Further diving into the evaluation metrics, Table 5 showcases the precision, recall, and F1-score values of the GBC classifier. Precision is the ratio of correctly predicted positive observations to the total predicted positives. Recall (Sensitivity) is the ratio of correctly predicted positive observations to all the observations in the actual class. The F1 Score is the weighted average of Precision and Recall. High and consistent recall values, as mentioned, signify that the classifier's performance remained stable and reliable throughout the evaluation, especially when assessed using ten-fold cross-validation. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample, and ten-fold cross-validation means the dataset was split into ten parts, with nine parts used for training and one part used for testing in each iteration.

To evaluate the quality of a prediction model, the Receiver Operating Characteristic (ROC) curve was plotted. It depicts true positive rates against false positive rates. Figure 3 displays this curve for the GBC classifier, showing its impressive area under a curve of 0.999 that provides further evidence for its superior quality prediction model.

VI. CONCLUSIONS

Early diagnosis of diabetes is vital to effective treatment and avoiding complications. In this study, we explored the early identification of diabetes and investigated its impact on classifier performance. At an early-stage diabetes risk prediction dataset from a diabetes hospital, our GBC algorithm provided an accuracy rate of 98.654%. Results also demonstrated the significant effect of normalisation techniques on classifier accuracy, with GBC reaching the highest accuracy of 99.038% when using Decimal Point Scaling, Z Score Normalisation, Max Normalisation, Maximum Absolute Scaling and Median and MAD Normalisation techniques. Therefore, our findings indicate that data mining algorithms can accurately detect early-stage diabetes without medical diagnosis; additionally, data normalisation techniques can further boost classifier performance.

REFERENCES

[1]. Kandhasamy, J.P.; Balamurali, *S. Performance Analysis of Classifier Models to Predict Diabetes Mellitus*. Procedia Comput. Sci. 2015, *47*, 45–51.

[2]. Yuvaraj, N.; SriPreethaa, *K.R. Diabetes prediction in healthcare systems using machine Learning algorithms on Hadoop cluster*. Clust. Comput. 2017, *22*, 1–9.

[3]. Tafa, Z.; Pervetica, N.; Karahoda, B. *An intelligent system for diabetes prediction*. In Proceedings of the 2015 4th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 14–18 June 2015; pp. 378–382.

[4]. Sisodia, D.; Sisodia, D.S. *Prediction of Diabetes using Classification Algorithms*. Procedia Comput. Sci. 2018, 132, 1578–1585.

[5]. Mercaldo, F.; Nardone, V.; Santone, *A. Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning Techniques*. Procedia Comput. Sci. 2017, 112, 2519–2528.

[6]. Negi, A.; Jaiswal, V. *A first attempt to develop a diabetes prediction method based on different global datasets*. In Proceedings of the 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 22–24 December 2016; pp. 237–241.

[7]. Olaniyi, E.O.; Adnan, K. *Onset diabetes diagnosis using artificial neural network.* Int. J. Sci. Eng. Res. 2014, *5*, 754–759.

[8]. Soltani, Z.; Jafarian, A. *A New Artificial Neural Networks Approach for Diagnosing Diabetes Disease Type II*. Int. J. Adv. Comput. Sci. Appl. 2016, 7, 89–94.

[9]. Somnath, R.; Suvojit, M.; Sanket, B.; Riyanka, K.; Priti, G.; Sayantan, M.; Subhas, B. *Prediction of Diabetes Type-II Using a Two-Class Neural Network*. In Proceedings of the 2017 International Conference on Computational Intelligence, Communications, and Business Analytics, Kolkata, India, 24–25 March 2017; pp. 65–71.

[10]. Mamuda, M.; Sathasivam, S. *Predicting the survival of diabetes using neural network.* In Proceedings of the AIP Conference Proceedings, Bydgoszcz, Poland, 9–11 May 2017; Volume 1870, pp. 40–46.

[11]. Vrushali R. Balpande and Rakhi D. Wajgi. 2017. *Prediction and severity estimation of diabetes using data mining technique*. 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 576-580.

[12]. Saranya C and Manikandan G. 2013. *A study on normalization techniques for privacy preserving data mining*. International Journal of Engineering and Technology (IJET) 5 (3), 2701-2704.

[13]. Zohair Ihsan, Mohd Yazid Idris, and Abdul Hanan Abdullah. 2013. *Attribute normalization techniques and performance of intrusion classifiers: A comparative analysis*. Life Science Journal 10 (4), 2568-2576.

[14]. RD Canlas. 2009. *Data mining in healthcare: Current applications and issues*. School of Information Systems & Management, Carnegie Mellon University, Australia.

AUTHOR

Ravikant Kholwal, a graduate from PDPM IIITDM Jabalpur, has significantly advanced intrusion detection systems, increasing accuracy by 25% through a novel algorithm. With publications in IJEAT and expertise in image classification, he has contributed to diverse projects using Django, ReactJs, and OpenCV. As a Software Engineer and CTO, he is proficient in JavaScript, React.js, AngularJS, Firebase, and is an AWS Certified Cloud Practitioner. Ravikant's blend of research and application skills distinguishes him in computing.