Original Article

Two Machine-learning Hybrid Models for Predicting Type 2 Diabetes Mellitus

Rahman Farnoosh, Karlo Abnoosian, Rasha Abbas Isewid

The School of Mathematics and Computer Science, Statistics, Iran University of Science and Technology, Tehran, Iran

Abstract

Background: The global increase in diabetes prevalence necessitates advanced diagnostic methods. Machine learning has shown promise in disease diagnosis, including diabetes. **Materials and Methods:** We used a dataset collected from the Medical City Hospital laboratory and the Specialized Center for Endocrinology and Diabetes at Al-Kindy Teaching Hospital in Iraq. This dataset includes 1000 physical examination samples from both male and female patients. The samples are categorized into three classes: diabetic (Y), nondiabetic (N), and predicted diabetic (P). The dataset contains twelve attributes and includes outlier data. Outliers in medical studies can result from unusual disease attributes. Therefore, consulting with a specialist physician to identify and handle these outliers using statistical methods is necessary. The main contribution of this study is the proposal of two hybrid models for diabetes diagnosis in two scenarios: (1) Scenario 1 (presence of outlier data): Hybrid Model 1 combines the K-medoids clustering algorithm with a Gaussian naive Bayes (GNB) classifier based on kernel density estimation (KDE) to handle outliers and (2) Scenario 2 (after removing outlier data): Hybrid Model 2 combines the K-means clustering algorithm with a GNB classifier based on KDE with suitable bandwidth. We performed principal component analysis to minimize dimensionality and evaluated the models using fivefold cross-validation. **Results:** All experiments were conducted in identical settings. Our proposed hybrid models demonstrated superior performance in two scenarios, handling and rejecting outliers, compared to other machine-learning models in this study, including support vector machines (with radial-based, polynomial, linear, and sigmoid kernel functions), decision trees (J48), and GNB classifiers for diabetes prediction. The average accuracy for Scenario 1 with Hybrid Model 1 was 0.9743, and for Scenario 2 with Hybrid Model 2, it was 0.9867. We also evaluated precision, sensitivity, and F1-score as performance metrics. **Conclusion:** This study presents two hybrid models for diabetes diagnosis, demonstrating high accuracy in distinguishing between diabetic and nondiabetic patients and effectively handling outliers. The findings highlight the potential of machine-learning techniques for improving the early diagnosis and treatment of diabetes.

Keywords: *Decision tree, diabetes mellitus prediction, Gaussian naive Bayes, kernel density estimation, K-means, K-medoids, support vector machine*

Submitted: 08-May-2024 **Revised:** 09-Sep-2024 **Accepted:** 22-Oct-2024 **Published:** 19-Apr-2025

Introduction

Type 2 diabetes is a prevalent and costly chronic disease associated with various health complications.[1] It affects the body's ability to use sugar (glucose) for energy and hampers proper insulin utilization, leading to elevated blood sugar levels if left untreated.[2,3] Symptoms may initially be mild and take years to manifest, often resulting in a late diagnosis when complications have already arisen.[4-6] Diabetes has reached epidemic proportions, affecting millions worldwide and placing a significant strain on healthcare systems. In 2022, 537 million adults were living with diabetes, and this number is projected to rise to 783 million by 2045. This global crisis demands urgent action, including promoting healthy lifestyles, ensuring early diagnosis, and providing effective treatment.[7,8] Machine-learning algorithms have been utilized in many fields, particularly medicine, to improve disease diagnosis.[9,10]

Kumar P *et al.* proposed a hybrid model combining bee colony algorithms and a fuzzy system for diabetes prediction.[11] Machine-learning algorithms aim to describe and predict data, assisting in the initial

How to cite this article: Farnoosh R, Abnoosian K, Isewid RA. Two machine-learning hybrid models for predicting type 2 diabetes mellitus. J Med Signals Sens 2025;15:11.

Address for correspondence:
Prof. Rahman Farnoosh, The School of Mathematics and Computer Science, Statistics, Iran University of Science and Technology, Tehran - 1684613114, Iran.
E-mail: rfarnoosh@iust.ac.ir

Access this article online

Website: www.jmssjournal.net

DOI: 10.4103/jmss.jmss_29_24

Quick Response Code:

assessment and diagnosis of diabetes.[12,13] Previous studies have explored various machine-learning techniques for predicting diabetes and evaluating their effectiveness and accuracy.[12,14] For example, Reinehr, Thomas, and Martin Wabitsch conducted a thorough investigation using a dataset of 520 cases and 17 features and found that the extra tree classifier (ETC) outperformed other algorithms, achieving an accuracy of 98.55%.[15] Another study compared decision tree (DT), support vector machine (SVM), and naive Bayes (NB) algorithms and concluded that NB exhibited the highest accuracy.[16] In addition, researchers in Luzhou, China, applied three machine-learning algorithms and found that random forests provided optimal accuracy for diabetes prediction.[17] They employed an area under the receiver operating characteristic (ROC) curve to measure the discriminatory capabilities of these models. Pal, Madhumita, and their colleagues proposed a machine-learning model for the early prediction of type 2 diabetes based on Indian diabetes data, in which they used the area under the ROC curve to examine and compare the performance of each model.[18] In their study, Zhou *et al*.[19] introduced a diabetes prediction model that uses Borota feature selection and ensemble learning. This model includes using Borota feature selection, extracting important features from the dataset, using the K-means++ algorithm for unsupervised data clustering, and adopting a cumulative ensemble learning approach for classification.

Given the severe complications and treatment costs associated with diabetes, there is a growing interest in novel approaches for its prevention and management. Medical studies may encounter unique cases or disease attributes, necessitating consultation with specialists, and consideration of data outliers.[20,21] The NB classifier has gained attention in medical diagnosis due to its simplicity and high accuracy on small datasets.[22,23] However, the assumption of a normal distribution in the Gaussian NB (GNB) classifier may limit its applicability. In this study, we examined a dataset that did not follow a normal distribution and sought alternative methods. Kernel density estimation (KDE) was employed as a nonparametric approach to estimate the probability density function without assuming the shape of the data distribution.[24,25] To address the limitations of KDE when data within a class originates from different distributions, we employed clustering methods to group data into smaller, more accurately classified subsets.

This article is structured as follows: Section 2 reviews the clustering methods used in this study. Section 3 covers classification methodologies. Section 4 describes the materials and methods, with a focus on Section 4.2, which details the proposed model. Section 5 explains the evaluation criteria for the models. Section 6 presents the test results. Section 7 discusses the results and suggests directions for future research. Finally, Section 8 concludes the study by summarizing the key findings and their implications for the field.

Clustering

Clustering is dividing the population or data points into several groups such that data points in the same group are more similar to other data points in the same group than to those in different groups. The aim is to segregate groups with similar traits and assign them into clusters.[26] In this study, we used two K-medoids clustering methods in the first proposed model in the presence of outliers and K-means in the second proposed model after outlier rejection, which is described below. All of these algorithms were implemented in the Python programming language environment.

K-means

K-means is a widely-used clustering algorithm that partitions a set X of points in a vector space into k clusters. The algorithm begins with an initial set of cluster centers, often chosen randomly from the data points, and iteratively assigns each point to the nearest centroid. After each assignment, the centroids are updated based on the mean of the points assigned to them. This process continues until convergence criteria, such as minimal change in centroids or a maximum number of iterations, are met.[27] For a comprehensive outline of the steps involved in this algorithm, please refer to Section A of the Appendix, where the corresponding pseudocode is provided.

K-medoids

The K-medoids algorithm is a clustering algorithm related to the K-means algorithm. Both the K-means and K-medoids algorithms are for partition (breaking the dataset up into groups). K-means attempts to minimize the total squared error, while K-medoids minimize the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the K-means algorithm, K-medoids choose data points as centers (medoids). A medoid can be a point in the data that acts as an example for other points in the.[28] This point is the central point in the cluster because it has the lowest average dissimilarity compared to the other data points in the cluster. K-means clustering algorithms related to outliers, dirty data, and abnormal data are highly sensitive.[29] This is because if a data point has large values, the data distribution may be biased,[30] but the K-medoids algorithm is more robust. For a detailed outline of the steps involved in this algorithm, please refer to Section A of the Appendix, where you can find the corresponding pseudocode.

Classification

In this study, we used NB, KDE, DT, and SVM classification methods. All of these classification algorithms were implemented in the Python programming language.

Naive Bayes

The NB classification algorithm for machine learning is a group of probabilistic classification methods based on the Bayesian theorem, and assumes that each attribute is conditionally independent of the other attributes.[31] Its advantages, such as easy understanding and simple implementation, have made it more useful than other machine-learning algorithms.[32-34]

Let $X = \{x_1, x_2,\ldots, x_N\}$ represents the set of training samples, where each i^{th} feature vector $X_i = [x_{i1}, x_{i2},\ldots, x_{id}]$. Let C be a set containing class labels (in this study, we have three classes), $P(x_1, x_2,\ldots, x_d|c)$ denotes the likelihood, P(c) and denotes the prior probability.[50] The classifier for selecting the most likely class by this algorithm is converted as follows:

$$\widehat{c_{NB}} = \underset{c\in C}{\operatorname{argmax}}\, P(c)\prod_{i=1}^{d} P(x_i|c) \quad (1)$$

For a comprehensive breakdown of the steps involved in this algorithm, along with the relevant equations, please refer to Section B of the Appendix, where you can find the corresponding pseudocode.[50]

Support vector machine

A SVM is a supervised learning method used in machine learning for classification. SVMs classify data by predicting labels from one or more attribute vectors, creating a decision boundary known as a hyperplane that separates two classes.[35] This hyperplane is determined by the closest data points from each class, known as support vectors.

In its simplest form, SVM does not support multiclass classification. Only binary classification and the separation of data points into two classes are supported. After breaking down a multiclass classification problem into numerous binary classification problems, the same method is applied to multiclass classification. The concept is to map the data points into a high-dimensional space to achieve a linear separation between the two classes once more. The one-versus-one (OvO) approach separates a multiclass classification problem into numerous binary classification problems and is known as the one-to-one approach.[36,37] For each pair of classes, a binary classifier is used.

The one-to-one classification strategy uses a $\frac{|C|(|C|-1)}{2}$ SVM, where |C| is the number of classes, which in this case is three. Figure 1a shows an example of a three-class classification problem with green, red, and blue classes. In the one-to-one technique, a hyperplane is required to separate every two classes, ignoring the points of the third class. This means that only the points of the two classes in the current split are considered in the separation. For example, the red-blue line [Figure 1b] aims to optimize the separation exclusively between the blue and red points. It has nothing to do with the green points.

Decision tree

One of the most common machine-learning models used for classification and regression is the DT model, which uses a divide-and-conquer approach to describe the classification process using a tree structure based on attributes.[38] The important components of a DT are nodes and branches. The most critical processes in creating a DT are splitting, pausing, and pruning. There are three types of nodes in a DT:

1. The decision node, also known as the root node, partitions all data into two or more mutually exclusive subsets
2. An internal node, also known as a chance node, represents one of the structural options available in a tree. The top edge is connected to the parent node, while the bottom edge is connected to the child or leaf nodes
3. The leaf or end node, which represents the final outcome, is the last type of node in the DT.

The final outcome is a decision. The results of the root node, or internal node, are represented by branches. A hierarchical structure with branches is used to create a DT. Each path from the root node to an internal node to a leaf node represents a classification decision rule. These paths can also be expressed using if-then rules.[39]

ID3, C4.5, J48, and other DT algorithms are common. The J48 DT method was employed in this investigation. This algorithm is a top-down divide-and-conquer strategy. For a comprehensive breakdown of the steps involved in this algorithm, along with the relevant equations, please refer to Section B of the Appendix, where you can find the corresponding pseudocode.

Kernel density estimation

KDE is a nonparametric density estimation method in which the probability density function (pdf) is estimated directly without a distribution assumption and based on only given data and similarity theory.[40] This method can be an effective way to estimate the data probability density function when we do not know the data distribution. For a mathematical representation of the method, see Section B of the Appendix.

Materials and Methods

Data

The Iraqi Patient Dataset for Diabetes[41] (IPDD) was obtained from 1000 individuals, including 565 males and 435 females aged 20-79 years, during in-hospital physical examinations at the Specialized Center for Endocrinology and Diabetes-Al-Kindy Teaching Hospital in Iraq. This dataset is divided into three regions: Diabetic (C0) with 844 samples, nondiabetic (C1) with 103 samples, and predicted diabetic (C2) with 53 samples. These included 12 physical examination indicators. Table 1 lists the attribute

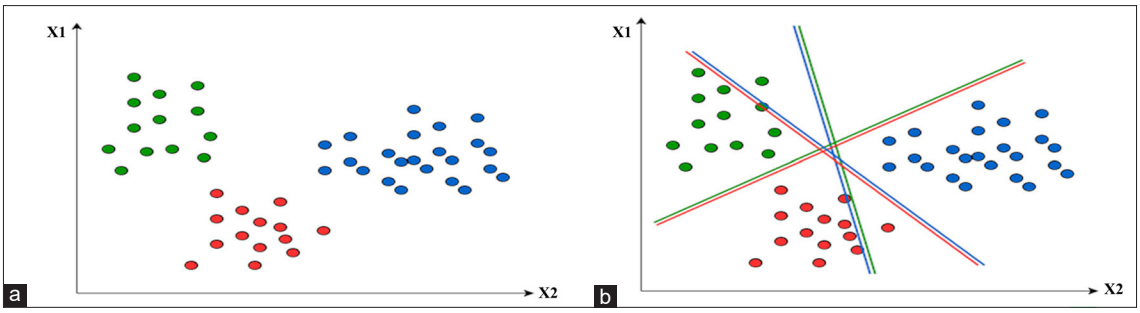

Figure 1: (a). An example of a classification problem of 3 classes: green, red, and blue. (b). The red–blue line is generated via a one-to-one technique to maximize the distance between the blue and red spots. Green points have nothing to do with it

Table 1: Overview of the Iraqi patient dataset for diabetes

Attributes	Description	Mean±SD
Gender	0 for females and 1 for male	0.565±0.4958
Age	Age in years	53.739±8.8557
FBS	Result of a blood sample taken after a patient fasted for at least 8 h	10.1443±5.0844
High BUN	BUN is the amount of urea nitrogen that is in your blood	5.1808±3.3486
Cr	Blood levels of chromium	69.28±62.2764
Chol	Fast Chol levels	4.9092±2.004
TG	Concentration tri glycoside levels	2.3506±1.3988
BMI	BMI	29.4255±4.8553
LDL	LDL	2.6145±1.1175
VLDL	VLDL	1.8573±3.6563
HDL	HDL	1.2067±0.6594
HbA1C	For the previous 2–3 months, average blood glucose (sugar) levels	8.2623±2.5370

FBS: Fasting blood sugar, BUN: Blood urea nitrogen, Cr: Chromium, HbA1C: Gyrated hemoglobin, BMI: Body mass index, LDL: Low-density lipoprotein, VLDL: Very LDL, HDL: High-density lipoprotein, Chol: Cholesterol, SD: Standard deviation

descriptions, and the distribution of each attribute in the dataset is shown in Figure 2, where the green, blue, and yellow color distributions denote the diabetic, nondiabetic, and predicted diabetic classes, respectively.

Proposed models

We will elaborate on each component of these two proposed models, as illustrated in Figures 3 and 4, in the following sections.

Data preprocessing

One of the important steps in machine-learning projects that are performed in the early stages is data preprocessing, which has an impact on the performance of the model. In this study, we performed the following steps to clean the data:

- Removing duplicate samples: After examining all 1000 data samples in this study, we found that seven of them were identical. Consequently, these duplicates were removed, leaving 993 samples
- Attribute conversion: The gender attribute values were changed to two values, 0 for females and 1 for males, and the class labels were 0 for Y, 1 for N, and 2 for P
- Filling in missing data: It is recommended to fill in missing or null values because they can lead to incorrect inferences for each class.[42] In this study, we use the K-nearest-neighbor (K-NN) algorithm to avoid the negative effect of missing data, and the results are shown in Figure 5
- Outlier rejection: In the first proposed model, we conducted classification on a dataset comprising 1000 samples, including outliers. Outliers are data points that deviate markedly from the general data distribution, potentially skewing the results of sensitive classifiers. Therefore, identifying and managing outliers is critical to ensuring the accuracy of our model.[43] We identified outliers in the dataset using an extended interquartile range (IQR) method. This method calculates the IQR as the difference between the third quartile (Q3) and the first quartile (Q1). In this study, we considered any data point falling below $Q1-\left(\frac{3}{2}\right)\times IQR$ or above $Q3+\left(\frac{3}{2}\right)\times IQR$ as an outlier. This extended range provides a more robust criterion for detecting outliers than the traditional IQR method. For example, we detected outliers in the low-density lipoprotein (LDL) attribute, as shown in Figure 6. To effectively manage and eliminate outliers, we implemented the OR(.) function, which utilizes this extended IQR range. By applying this function, we systematically excluded identified outliers from the dataset. This approach mitigates the influence of extreme values on our classification model, thereby enhancing the robustness and accuracy of our predictions. Detailed information about the OR(.) function, including the specific formula, can be found in Section C of the Appendix
- Normalization: Because the values of some attributes in the data have a wide range, this can have a serious impact on the performance of the classifier. To rescale the range of our continuous features to an interval between 0 and 1, we employed min-max

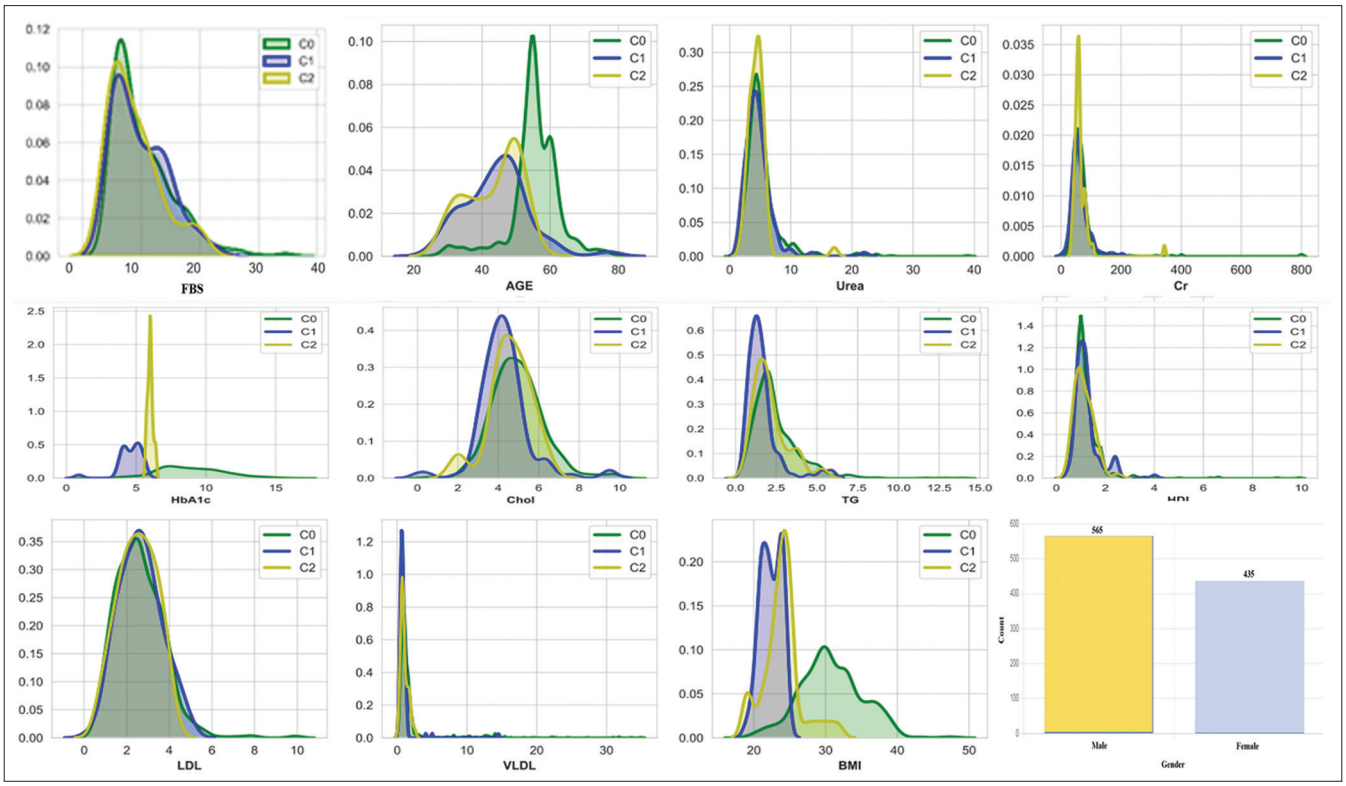

Figure 2: The Iraqi Patient Dataset for Diabetes dataset population distribution of all attributes, where the green, blue, and yellow color distributions indicate diabetic (C0) individuals, nondiabetic (C1) individuals, and predicted diabetic (C2) individuals, respectively

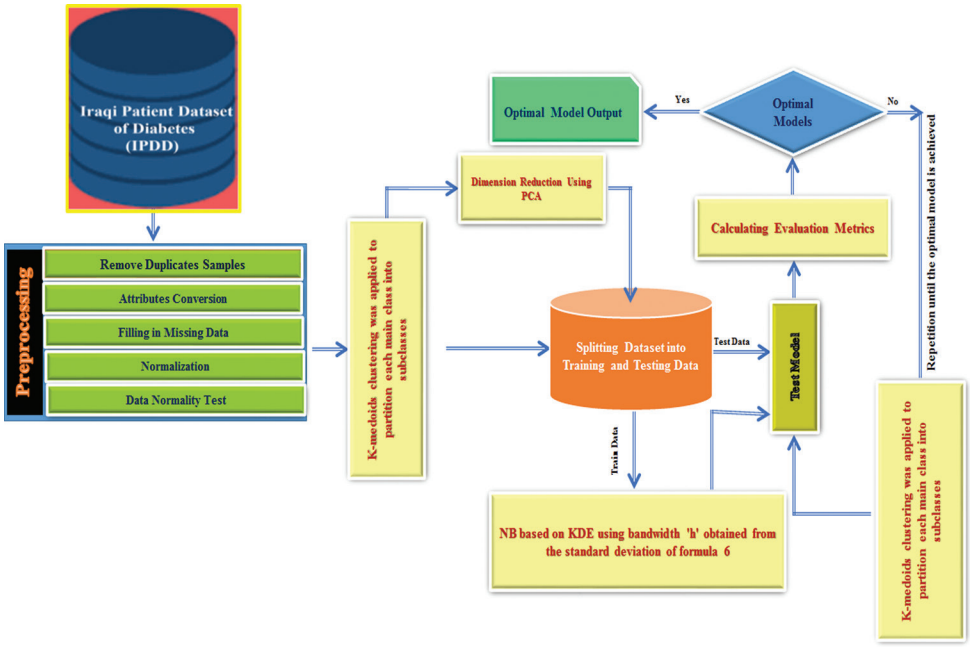

Figure 3: M1. In the first scenario, which involves the presence of outliers. Thus, we used K-medoids clustering methods and then naive Bayes-based Kernel density estimation

normalization.[44] For a more in-depth explanation of this technique, please refer to Section C of the Appendix

- Data normality test: After the normalization stage in this study, we used various methods for testing data normality, such as the D'Agostino K-square test, Anderson–Darling test, Shapiro–Wilk's test with a hypothesis test, and finally, the quantile-quantile plot for more certainty. We concluded that some attributes could not have a normal distribution

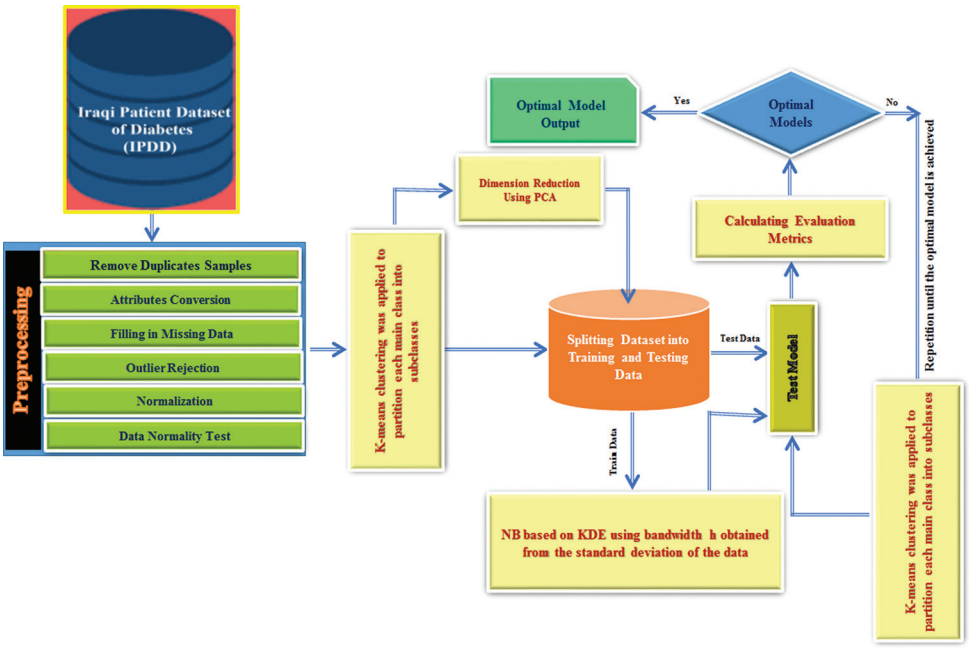

Figure 4: M2. In the second scenario, after outlier rejection, we used K-means clustering and then naive Bayes-based kernel density estimation

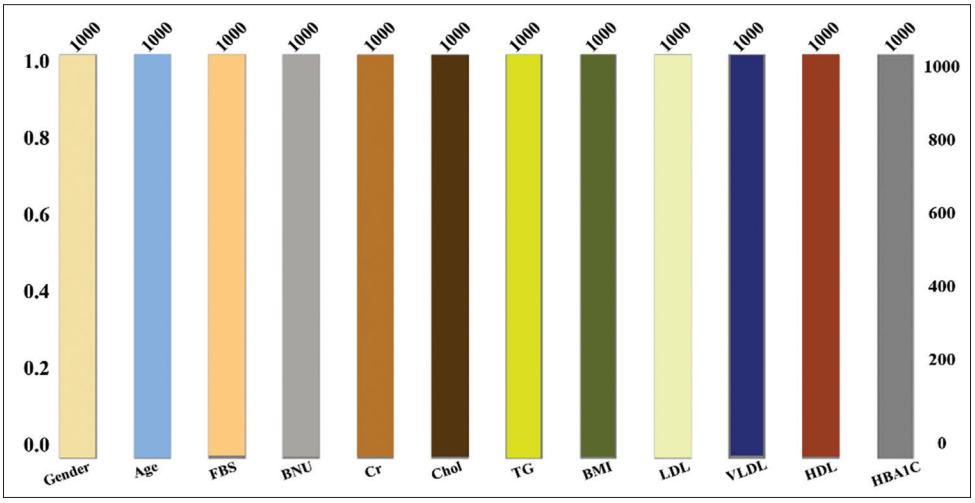

Figure 5: Results from using the K-nearest-neighbor algorithm to fill in missing and empty data

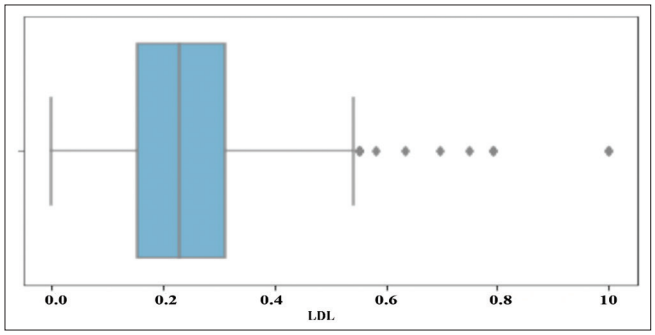

Figure 6: Outliers associated with low-density lipoprotein

- Finally, the dataset was randomly divided into two parts: 1/5 for testing and 4/5 for training the models.

Clustering and classification in the proposed models

Considering the various tests performed in the data normality test section in the data preprocessing section, we concluded that some attributes lack a normal distribution, so the use of an NB classifier reduces the performance of the classifier; thus, we used KDE. For example, for the AGE and TG attributes in Figure 7a and b, we see that using KDE works better than the normal distribution; however, according to Figure 7a and b, we see that the data and attributes of a class lack a unique distribution and have different distributions in different regions; we use a clustering method to divide the data of a class into several subclasses with the same statistical similarity in each cluster. According to the results we obtained in the

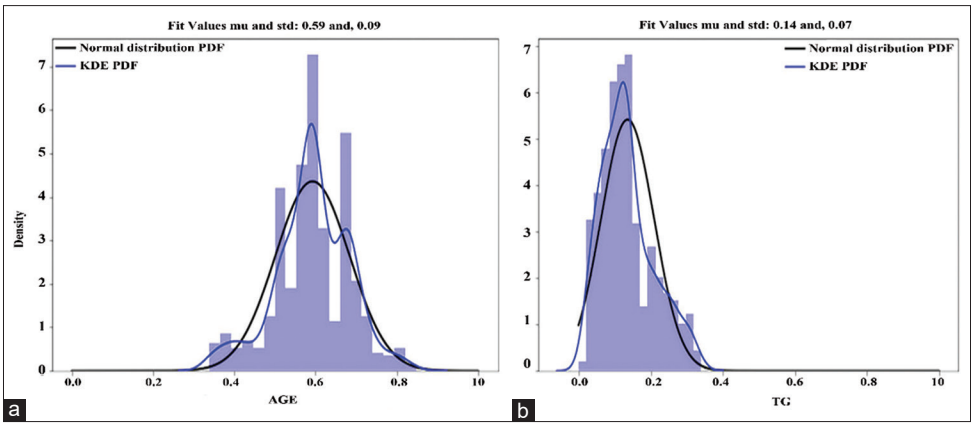

Figure 7: (a) The Kernel density estimation performance is better than that of the normal distribution. For the AGE attribute and (b) for the TG attribute. In addition, we see that both attributes have different distributions in different regions. In addition, we used a clustering method to divide the data of a class into several clusters with the same statistical similarity in each cluster

previous sections, because the K-means clustering method is sensitive to outliers, this clustering algorithm cannot be used in the first scenario (M1), which is the presence of an outlier. Thus, we used K-medoids clustering for M1 macrophages. In the second scenario (M2), after outlier rejection, we used K-means clustering [Figures 3 and 4].

Suppose we have N random variables X = [X_1,X_2, .., X_n] belonging to class c, drawn from a multimodal mixed-density function. in such a way that it is a multimodal random variable. Each main class has several subclasses. Samples X^1,X^2, ..., X^n belongs to subclasses X^1,X^2, ..., X^n, where $X^i = [X^i_1,\ldots,X^i_{N_i}]$ and $N = \sum_{i=1}^{N} N_i$.

Now, with the entry of a new observation X, the conditional probability is obtained with the finite model mixed with the following equation:

$$p(c_i \mid x) = \sum_{k=1}^{n_i} p(c_i^k \mid x) = \sum_{k=1}^{n_i} p(c_i^k) \prod_{j=1}^{r} p(x_j \mid C_i^k) \quad (2)$$

where the prior probability for each subclass is calculated as follows:

$$P(c_i^k) = \frac{N_i^k}{N_i} \quad (3)$$

The likelihood probability for each subclass k-th is from the main class i-th calculated using KDE as follows:

$$\left[p(x_j \mid c_i^k)\right]_{k=1,2,3,..,n_i} = \frac{1}{n_i h_{i,j}^k} \sum_{t=1}^{N_i^k} K\left(\frac{x_j - x_{i,j,t}^k}{h}\right) \quad (4)$$

N_i^k and $h_{i,j}^k$ are the number of samples and the calculated bandwidth for the attribute of the data belonging to the k subclass of the i-th main class, respectively; r is the number of attributes of each sample; c_i^k represents the label of the k-th subclass of the i-th main class; and n_i indicates the number of i-th subclass of the main class. In this study, we employed 6 different types of kernel functions (K1 to K6), which are listed in Section D of the Appendix and Table 2.

Table 2: The values of bandwidth h used for comparison

Bandwidth	Values of bandwidth
h_1	Used Equation 5 based on the SD of Equation 6
$h_2 = \frac{h_1}{2}$	Used Equation 5 based on the SD of Equation 6
$h_3 = \frac{2h_1}{5}$	Used Equation 5 based on the SD of Equation 6
$h_4 = \frac{2h_1}{5}$	Used Equation 5 based on the SD of Equation 6
h_5	Used Equation 5 based on the SD estimated from data samples
$h_6 = \frac{h_5}{2}$	Used Equation 5 based on the SD estimated from data samples
$h_7 = \frac{3h_5}{2}$	Used Equation 5 based on the SD estimated from data samples
$h_8 = \frac{2h_5}{5}$	Used Equation 5 based on the SD estimated from data samples

SD: Standard deviation

Another important factor influencing the quality of the probability density function $\hat{f}$ using KDE is the smoothing parameter, or bandwidth, and an improper h bandwidth may result in undersmoothing or oversmoothing. One of the most methods for selecting bandwidths is Silverman's rule of thumb (ROT) method.[40] The optimal bandwidth for the Gaussian kernel function is defined as follows:

$$h_{i,j}^k = \widehat{\sigma_{i,j}^k} \left(\frac{4}{(r+2)N_i^k}\right)^{\frac{1}{r+1}} \quad (5)$$

where $\widehat{\sigma_{i,j}^k}$ is the estimated standard deviation of the j-th attribute of subclass k-th from the main class i-th. It performs well even when the probability density function is approximately not similar in shape to the Gaussian function.[45] The standard deviation is sensitive to outliers. Outliers can increase the standard deviation, misrepresent

the dispersion, and select the wrong bandwidth value. Silverman's ROT uses standard deviation. Because the standard deviation is sensitive to outliers, we use a strong, changed standard deviation as follows.[46]

$$\widehat{\sigma_{i,j}^{k}} = \frac{\text{Median}\left(\left|X_j - \text{Median}\left(X_j\right)\right|\right)}{0.6745} \quad (6)$$

In the last equation, the median is used, which is more resistant to outlying data than the mean. The scaling factor of 0.6745 used in the equation is used to obtain the correct answer for data that follow a Gaussian distribution. In the normal distribution with mean μ and standard deviation σ, the probability that a value is within one standard deviation from the mean is approximately 0.6745. Using this scaling factor, estimates obtained by robust statistical methods are usually adapted to the correct approximate distribution for data that do not follow a normal distribution. In fact, the use of this scaling factor leads to robust statistical estimates for the parameters of sparse data and distributions that differ from the normal distribution.

The selection of the appropriate kernel function is a crucial factor in the accuracy of the probability density function estimation via KDE.

In this study, in M1, which is made in the presence of outliers, we used the standard deviation that estimated each attribute in the ROT rule, and in M2, which is the outlier rejection process, we used the standard deviation that estimated each attribute in the ROT rule. All bandwidth used in this study is listed in Table 2. The pseudocode of the proposed models is given in section E of the Appendix.

Feature extraction

One way to improve computational efficiency and increase the reliability of estimated joint probabilities is to find useless features that have little effect on classification and impair predictive performance. We used the principal component analysis (PCA) method to reduce the maximum relevance and dimensionality. PCA[47] is a mathematical approach for reducing the dimensionality of data based on identifying directions called principal components and retaining the most variation in the dataset. Using a few components, each sample can be represented by a relatively small number instead of many variables.

Evaluation metrics

In this study, to evaluate the effectiveness of the multiclass classification model, we employed average accuracy (AAC), precision, recall, and F1-score metrics.[48,49] The formulas for these metrics are provided in Section G of the Appendix.

Results

In this section, we implement the proposed models [Figures 3 and 4] based on different kernel functions using different algorithms and bandwidths.

We then applied these models to the Iraqi diabetic patient dataset, which includes 12 features. Finally, on the basis of different evaluation metrics, we measured the multiclass classification performance. To make a more accurate comparison, we initially used all the features to predict diabetes, and the results are presented in Tables 3 and 4. We concluded that in the first scenario, which is in the presence of outliers [Table 3], M1 had better results than the other models. In the second scenario, which is after the outlier rejection process [Table 4], M2 had better results than the other models based on the results obtained from Tables 3 and 4.

We found that glucose had the greatest information gain, confirming common sense and providing the foundation for clinical diagnosis. However, there were diabetic patients in the dataset who had a fasting blood sugar of less than 6.8. We reasoned that this could be because the patients had been injected with insulin before the physical examination to manage their blood sugar. Tables 5 and 6 show the results of using all of the attributes without blood glucose to predict diabetes to better understand the value of other indices in predicting diabetes. Table 5 shows that in the presence of outliers, M1 had the optimal results, whereas Table 6 shows that after the outlier rejection procedure, M2 had the optimal results.

Features were then reduced using PCA. According to the Kaiser–Meyer–Olkin and Bartlett's tests, the dataset was suitable for PCA to reduce the number of features. We also obtained the eigenvalues and composition matrix. Finally, we extracted eight new features for this dataset from the composition matrix and interpreted the total variance. We then conducted an experiment using the additional features, and the results are presented in Tables 7 and 8 for the first and second scenarios, respectively. The results show that the AAC dataset is superior to the previously mentioned approaches. The results show that PCA is suitable for this dataset.

The results of the above experiments are shown in Figures 8-10. In terms of the accuracy of each method, we can summarize that these results provide a better comparison between different models.

When we employed all of the data to predict diabetes, we discovered that the two proposed models outperformed the other models in both the first and second scenario situations: In the first scenario in the presence of outliers and the second scenario following the outlier rejection process [Figure 8].

From Figure 8, with 12 or all the features without blood glucose features, when the presence of outliers is taken into account, the first proposed models of SVM (polynomial), GNB, and J48 have similar performances, and the second proposed model is better. After the outlier rejection process, the NB, DT (J48), and SVM (polynomial) models have

Table 3: Predict diabetes by using all features for the first scenario

	AAC		PPV_M		TPR_M		$F_{1,M}$	
	M1	**M2**	**M1**	**M2**	**M1**	**M2**	**M1**	**M2**
h_1								
K1	0.9400±0.0020	0.9199±0.0013	0.5101±0.0017	0.6273±0.0090	0.6327±0.0120	0.4054±0.0100	0.5648±0.0150	0.4925±0.0200
K2	0.9433±0.0012	0.9100±0.0015	0.9278±0.0012	0.6273±0.017	0.7591±0.002	0.4054±0.0015	0.9231±0.0018	0.4925±0.0025
K3	0.9300±0.0005	0.9100±0.0004	0.9127±0.0008	0.6273±0.0007	0.6380±0.0014	0.4054±0.0012	0.7510±0.0013	0.4925±0.0017
K4	0.9233±0.0010	0.9100±0.0010	0.7303±0.0012	0.6273±0.0018	0.4425±0.0008	0.4054±0.0004	0.5511±0.0015	0.4925±0.0019
K5	0.9435±0.0015	0.9300±0.0018	0.8847±0.0017	0.5425±0.0023	0.6989±0.0017	0.5108±0.0008	0.7809±0.0003	0.5262±0.0009
K6	0.9500±0.0012	0.9199±0.0025	0.8879±0.0012	0.6273±0.0019	0.6404±0.0023	0.4054±0.0017	0.7441±0.0025	0.4925±0.0029
h_2								
K1	0.9400±0.0022	0.9300±0.0018	0.5101±0.0012	0.9127±0.0023	0.6327±0.0028	0.6380±0.0025	0.5648±0.0014	0.7510±0.0019
K2	0.9433±0.0013	0.9199±0.0012	0.5527±0.0017	0.6273±0.0024	0.6014±0.0026	0.4054±0.0014	0.5760±0.0018	0.4925±0.0017
K3	0.9333±0.0009	0.9128±0.0008	0.5634±0.0009	0.7740±0.0002	0.4692±0.007	0.4715±0.0003	0.5394±0.0007	0.7119±0.008
K4	0.9400±0.0030	0.9199±0.0004	0.5101±0.0004	0.6273±0.0009	0.6320±0.0004	0.4054±0.0006	0.5648±0.0008	0.4925±0.0006
K5	0.9433±0.0018	0.9300±0.0007	0.5527±0.0008	0.5425±0.0005	0.6014±0.0007	0.5108±0.0002	0.5760±0.0004	0.5262±0.0007
K6	0.9435±0.0013	0.9233±0.0017	0.9009±0.0018	0.4803±0.0023	0.5885±0.0033	0.6230±0.0035	0.7119±0.0018	0.5425±0.0020
h_3								
K1	0.9300±0.0019	0.9100±0.0022	0.5425±0.0028	0.4765±0.0027	0.5108±0.0032	0.6211±0.0026	0.5262±0.0014	0.5393±0.0016
K2	0.9400±0.0020	0.9199±0.0008	0.5101±0.0018	0.6273±0.0019	0.6327±0.0007	0.4054±0.0009	0.5648±0.0023	0.4925±0.0029
K3	0.9400±0.0010	0.9199±0.0009	0.5101±0.0011	0.6273±0.0019	0.6327±0.0018	0.4054±0.0024	0.5648±0.0028	0.4925±0.0019
K4	0.9400±0.0025	0.9199±0.0026	0.5101±0.0042	0.6273±0.0041	0.6327±0.0012	0.4054±0.0016	0.5648±0.0016	0.4925±0.0019
K5	0.9466±0.0009	0.9433±0.0019	0.9038±0.0031	0.7740±0.0018	0.6219±0.0020	0.6327±0.0012	0.7368±0.0003	0.5760±0.00023
K6	0.9500±0.0009	0.9199±0.00010	0.8879±0.0023	0.6273±0.0033	0.6404±0.0025	0.4054±0.0042	0.7441±0.0043	0.4925±0.0041
h_4								
K1	0.9400±0.0012	0.9300±0.0008	0.5101±0.0007	0.5425±0.0008	0.6327±0.0002	0.5108±0.0001	0.5648±0.0007	0.5267±0.0009
K2	0.9487±0.0018	0.9400±0.001	0.9313±0.003	0.6418±0.0009	0.7554±0.0008	0.5526±0.0007	0.8341±0.0001	0.5939±0.0003
K3	0.9450±0.0009	0.9199±0.0008	0.8879±0.0009	0.6273±0.0004	0.6404±0.0003	0.4054±0.0007	0.7441±0.0004	0.4925±0.0002
K4	0.9600±0.0012	0.9199±0.0036	0.8962±0.0018	0.6273±0.0019	0.7164±0.0031	0.4054±0.0035	0.7759±0.0023	0.4925±0.0028
K5	0.9567±0.0015	0.9233±0.0034	0.5053±0.0016	0.4803±0.0014	0.6317±0.0013	0.6230±0.0023	0.5511±0.0025	0.9073±0.0014
K6	0.9600±0.0025	0.9400±0.0023	0.9299±0.0040	0.5101±0.0033	0.7795±0.0028	0.6327±0.0018	0.7856±0.0031	0.5648±0.0031
GNB	0.9487±0.0015		0.9100±0.0025		0.5526±0.0016		0.5939±0.0003	
J48	0.9466±0.0012		0.9038±0.0012		0.6219±0.0009		0.7368±0.0036	
SVM (RBF)	0.9128±0.0016		0.7740±0.0009		0.4715±0.0012		0.5860±0.0041	
SVM (ploynomial)	0.9434±0.0032		0.9009±0.0008		0.5885±0.0004		0.7119±0.0014	
SVM (linear)	0.9333±0.0021		0.6344±0.0019		0.4692±0.0036		0.5394±0.0004	
SVM (sigmoid)	0.9033±0.0033		0.6232±0.0003		0.3846±0.0008		0.4756±0.0023	

SVM – Support vector machine; GNB – Gaussian NB; TPR – True positive rate; RBF – Radial basis function; PPV – Positive predictive value; AAC – Average accuracy

Table 4: Predict diabetes by using all features for the second scenario

	AAC		PPV_M		TPR_M		$F_{1,M}$	
	M1	**M2**	**M1**	**M2**	**M1**	**M2**	**M1**	**M2**
h_1								
K1	0.9435±0.0012	0.9687±0.0011	0.8847±0.0016	0.8111±0.0004	0.6989±0.0033	0.6233±0.0011	0.7809±0.0004	0.7049±0.0007
K2	0.9589±0.0013	0.9641±0.0025	0.9406±0.00030	0.9168±0.0018	0.7081±0.0035	0.7689±0.0033	0.8080±0.0004	0.8364±0.0034
K3	0.9589±0.0003	0.9641±0.0035	0.9406±0.0022	0.9168±0.0016	0.7081±0.0018	0.7689±0.0028	0.8080±0.0011	0.8364±0.0025
K4	0.9589±0.0008	0.9641±0.0025	0.9406±0.0018	0.9168±0.0004	0.7081±0.0004	0.7689±0.0011	0.8080±0.0035	0.8364±0.0034
K5	0.9435±0.0014	0.9692±0.0035	0.8847±0.00030	0.9197±0.0018	0.6989±0.0028	0.8244±0.0018	0.7809±0.0033	0.8695±0.0028
K6	0.9589±0.00041	0.9641±0.0018	0.9406±0.0011	0.9168±0.0033	0.7081±0.0004	0.7689±0.0025	0.8080±0.0035	0.8364±0.0011
h_2								
K1	0.95890.00032	0.9641±0.0007	0.9406±0.0035	0.9168±0.0042	0.7081±0.0031	0.7689±0.0042	0.8080±0.00030	0.8364±0.0011
K2	0.9589±0.00021	0.9692±0.0035	0.9406±0.0004	0.9831±0.0018	0.7081±0.0016	0.8244±0.0028	0.8080±0.0034	0.8880±0.0031
K3	0.9589±0.0012	0.9692±0.0007	0.9406±0.0035	0.9831±0.0004	0.7081±0.00030	0.8244±0.0042	0.8080±0.0033	0.8880±0.0007
K4	0.9589±0.0034	0.9692±0.0028	0.9406±0.0011	0.9831±0.0022	0.7081±0.0018	0.8244±0.0021	0.8080±0.0035	0.8880±0.0034
K5	0.9538±0.0004	0.9692±0.0042	0.9001±0.00030	0.9831±0.0031	0.7051±0.0030	0.8244±0.0031	0.7907±0.0031	0.8880±0.0025
K6	0.9589±0.0034	0.9692±0.0007	0.9406±0.0034	0.9831±0.0035	0.7081±0.0042	0.8244±0.0031	0.8080±0.00029	0.8880±0.000723
h_3								
K1	0.9435±0.0004	0.9487±0.0036	0.8847±0.0028	0.8111±0.0033	0.6989±0.0018	0.6233±0.0035	0.7907±0.0035	0.7049±0.00030
K2	0.9589±0.0035	0.9641±0.0025	0.9406±0.00030	0.9168±0.0018	0.7081±0.0004	0.7689±0.0011	0.8080±0.0033	0.8364±0.0034
K3	0.9589±0.0011	0.9641±0.0028	0.9406±0.0034	0.9168±0.0022	0.7081±0.0018	0.7689±0.00030	0.8080±0.0035	0.8364±0.0042
K4	0.9589±0.0035	0.9641±0.0004	0.9406±0.0028	0.9168±0.0014	0.7081±0.0016	0.7689±0.0035	0.8080±0.0028	0.8364±0.0025
K5	0.9538±0.0032	0.9692±0.0018	0.9001±0.00030	0.9831±0.0025	0.7051±0.0042	0.7930±0.0004	0.7907±0.0025	0.8880±0.0031
K6	0.9100±0.0035	0.9692±0.0033	0.6256±0.0019	0.9831±0.0018	0.3686±0.0007	0.7930±0.0031	0.4639±0.0033	0.8880±0.0033
h_4								
K1	0.9538±0.0035	0.9641±0.0025	0.9001±0.0028	0.9168±0.0004	0.7051±0.0034	0.7689±0.0022	0.7907±0.0031	0.8364±0.0036
K2	0.9641±0.0004	0.9692±0.0031	0.9568±0.0011	0.9831±0.0011	0.7588±0.0033	0.7930±0.0018	0.8464±0.0031	0.8880±0.0042
K3	0.9641±0.0031	0.9692±0.0035	0.9568±0.0042	0.9831±0.0031	0.7588±0.0034	0.7930±0.0031	0.8464±0.0035	0.8880±0.0028
K4	0.96411±0.0031	0.9692±0.0028	0.9568±0.0031	0.9831±0.0035	0.7588±0.0042	0.7930±0.0028	0.8464±0.0031	0.8880±0.0018
K5	0.9589±0.0007	0.9692±0.0025	0.9139±0.0004	0.9831±0.0022	0.7081±0.0033	0.7930±0.0028	0.7979±0.0007	0.8880±0.0027
K6	0.9641±0.00030	0.9692.±0.0014	0.956±0.00288	0.9831±0.0042	0.7588±0.0025	0.7930±0.0018	0.8464±0.0034	0.8880±0.0004
GNB	0.9500±0.0004		0.8879±0.0011		0.6404±0.0033		0.9100±0.0007	
J48	0.9533±0.0035		0.8867±0.0021		0.6696±0.0031		0.9102±0.0035	
SVM (RBF)	0.9199±0.0011		0.5731±0.0014		0.4220±0.0022		0.4861±0.0004	
SVM (ploynomial)	0.9533±0.0025		0.8867±0.0016		0.6692		0.8425	
SVM (linear)	0.9435±0.00030		0.8847±0.0033		0.6989±0.0011		0.8387±0.0018	
SVM (sigmoid)	0.9333±0.0004		0.9222±0.0018		0.6682±0.0007		0.7749±0.0025	

SVM – Support vector machine; GNB – Gaussian NB; TPR – True positive rate; RBF – Radial basis function; PPV – Positive predictive value; AAC – Average accuracy

Table 5: Without utilizing blood glucose, predict diabetes using all features glucose for the first scenario

	AAC		PPV_M		TPR_M		$F_{1,M}$	
	M1	**M2**	**M1**	**M2**	**M1**	**M2**	**M1**	**M2**
h_1								
K1	0.9333±0.0012	0.9333±0.0015	0.8465±0.0023	0.8465±0.0009	0.7405±0.0019	0.7405±0.0016	0.8065±0.0011	0.4407±0.0033
K2	0.9333±0.0014	0.8900±0.00030	0.8465±0.0033	0.3441±0.0011	0.7405±0.0025	0.6393±0.0014	0.8065±0.0022	0.4407±0.0033
K3	0.9333±0.0025	0.8900±0.0004	0.8465±0.0028	0.3441±0.0033	0.7405±0.0022	0.6393±0.00030	0.8065±0.0018	0.4407±0.0035
K4	0.9333±0.0035	0.8900±0.0018	0.8465±0.0004	0.3441±0.0019	0.7405±0.0023	0.6393±0.0025	0.8065±0.0035	0.4407±0.0007
K5	0.9333±0.0011	0.8900±0.0023	0.8465±0.00030	0.5216±0.0015	0.7405±0.0018	0.7405±0.0016	0.8065±0.0009	0.0550±0.0028
K6	0.9300±0.0015	0.8900±0.0009	0.6127±0.0012	0.3441±0.0004	0.8394±0.0022	0.6393±0.0033	0.7258±0.0033	0.4407±0.0007
h_2								
K1	0.9300±0.0035	0.8900±0.0004	0.6127±0.0011	0.3441±0.0025	0.8394±0.00030	0.6393±0.0033	0.7258±0.0016	**0.4407±0.0028**
K2	0.9333±0.0014	0.9100±0.0018	0.7633±0.0004	0.8693±0.0011	0.8400±0.00030	0.7598±0.0011	0.8185±0.0033	**0.8110±0.0033**
K3	0.9333±0.0028	0.8733±0.0033	0.7633±0.0009	0.5522±0.0004	0.7598±0.0025	0.5073±0.0018	0.8109±0.0033	**0.5288±0.0028**
K4	0.9333±0.0018	0.9333±0.00030	0.7633±0.0016	0.6229±0.0014	0.8400±0.0009	0.4615±0.0015	0.7100±0.0011	**0.8109±0.0028**
K5	0.9400±0.0022	0.9333±0.0023	0.7633±0.0004	0.5940±0.00030	0.5108±0.0033	0.6693±0.0023	0.5492±0.0025	**0.7614±0.0033**
K6	0.9423±0.0014	0.8966±0.0012	0.7633±0.0004	0.6307±0.0009	0.7598±0.0019	0.3697±0.0018	0.7901±0.0015	**0.46620.0012**
h_3								
K1	0.9400±0.00030	0.9366±0.0007	0.7633±0.0018	0.640±0.000300	0.6893±0.0025	0.6400±0.0028	0.7614±0.0033	0.5591±0.0035
K2	0.9400±0.0025	0.8966±0.0012	0.7633±0.0016	0.6307±0.0015	0.6978±0.0033	0.3697±0.0023	0.8179±0.0033	0.4662±0.0018
K3	0.9400±0.0009	0.8966±0.0016	0.7633±0.0023	0.6307±0.0011	0.6978±0.0012	0.3697±0.0015	0.8179±0.00030	0.4662±0.0011
K4	0.9333±0.0018	0.9066±0.0022	0.7633±0.0015	0.6200±0.0016	0.7598±0.00030	0.4082±0.0004	0.8010±0.0025	0.4923±0.0035
K5	0.9333±0.0011	0.8733±0.0028	0.7633±0.0004	0.5522±0.0033	0.8400±0.0009	0.5073±0.0022	0.8185±0.0018	0.5288±0.0019
K6	0.9423±0.0035	0.8966±0.0014	0.7633±0.00030	0.6307±0.0025	0.7598±0.0033	0.3697±0.0018	0.8109±0.0028	0.4662±0.0016
h_4								
K1	0.9589±0.0022	0.9333±0.0011	0.8500±0.0016	0.6229±0.0012	0.8878±0.0018	0.4615±0.0014	0.8101±0.0025	0.5325±0.0004
K2	0.9589v	0.9333±0.0018	0.8500±0.0015	0.6322±0.0016	0.8878	0.3846±0.0007	0.8101±0.0028	0.4756±0.0019
K3	0.9589±0.0035	0.9000±0.0028	0.8500±0.0033	0.6595±0.0018	0.8878±0.0016	0.4174±0.0025	0.8101±0.0035	0.5102±0.0042
K4	0.9500±0.0022	0.9100±0.0009	0.9006±0.0035	0.6215±0.0019	0.7491±0.0011	0.4210±0.00030	0.8179±0.0028	0.5020±0.0018
K5	0.9400±0.0011	0.9333±0.0015	0.9870±0.0033	0.622±0.00169	0.6978±0.0028	0.4615±0.0015	0.7850±0.0022	0.5325±0.0004
K6	0.9500±0.0014	0.8966±0.0042	0.9006±0.0004	0.3707±0.0009	0.7491±0.0025	0.3697±0.0018	0.8179±0.0035	0.4662±0.0007
GNB	0.9433		0.8693		0.7598		0.8109	
J48	0.9400±0.0023		0.8970±0.0012		0.6879±0.0011		0.8179	
SVM (RBF)	0.8900±0.0009		0.6127±0.0033		0.3441±0.0004		0.4407±0.0022	
SVM (ploynomial)	0.9400±0.0016		0.8970±0.0019		0.6978±0.00030		0.8179±0.0025	
SVM (linear)	0.9300±0.0014		0.5425±0.0011		0.5108±0.0011		0.5262±0.0018	
SVM (sigmoid)	0.8966±0.0015		0.6307±0.0015		0.6397±0.0009		0.4662±0.0012	

SVM – Support vector machine; GNB – Gaussian NB; TPR – True positive rate; RBF – Radial basis function; PPV – Positive predictive value; AAC – Average accuracy

Table 6: Without utilizing blood glucose, predict diabetes using all features for the second scenario

	AAC		PPV_M		TPR_M		$F_{1,M}$	
	M1	**M2**	**M1**	**M2**	**M1**	**M2**	**M1**	**M2**
h_5								
K1	0.9128±0.0033	0.9435±0.0019	0.7740±0.0009	0.9009±0.0004	0.4715±0.0023	0.5885±0.0016	0.5860±0.0015	0.7119±0.0011
K2	0.9128±0.0016	0.9435±0.0025	0.7740±0.0033	0.9009±0.0018	0.4715±0.0022	0.5885±0.0028	0.5860±0.0007	0.7119±0.0004
K3	0.9128±0.0028	0.9435±0.0014	0.7740±0.0025	0.9009±0.0009	0.4715±0.0023	0.5885±0.0033	0.5860±0.00030	0.7119±0.0033
K4	0.9128±0.0025	0.9435±0.0011	0.7740±0.0015	0.9009±0.0004	0.4715±0.00030	0.5885±0.0011	0.5860±0.0016	0.7119±0.0004
K5	0.9076±0.0015	0.9435±0.0014	0.7558±0.0004	0.9009±0.0012	0.4715±0.0022	0.588±0.00115	0.5785±0.0018	0.7119±0.0009
K6	0.9128±0.0011	0.9487±0.0028	0.7740±0.0012	0.6418±0.0025	0.4715±0.0019	0.5526±0.0014	0.5860±0.0018	0.5539±0.0016
h_6								
K1	0.9128±0.0004	0.9487±0.0011	0.7740±0.0009	0.6418±0.0004	0.4715±0.00030	0.5526±0.0035	0.5860±0.0042	0.5539±0.0007
K2	0.9128±0.0033	0.9589±0.0014	0.7740±0.0018	0.7816±0.0019	0.4715±0.0023	0.8878±0.0004	0.5860±0.0016	0.5539±0.0016
K3	0.9128±0.0014	0.9487±0.0015	0.7740±0.0016	0.6418±0.0009	0.4715±0.0004	0.5526±0.0033	0.5860±0.0022	0.5539±0.0025
K4	0.9128±0.0028	0.9487±0.0035	0.7740±0.0018	0.6418±0.0028	0.4715±0.0007	0.5526±0.0016	0.5860±0.0035	0.5539±0.0042
K5	0.9076±0.0022	0.9487±0.0015	0.7558±0.00030	0.6418±0.0035	0.4715±0.0023	0.5526±0.0011	0.5785±0.0004	0.5539±0.0025
K6	0.9128±0.0007	0.9487±0.0028	0.7740±0.0042	0.6418±0.0018	0.4715±0.0035	0.5526±0.0031	0.5860±0.0031	0.5539±0.0028
h7								
K1	0.9128	0.9487	0.7740	0.6418	0.4715	0.5526	0.5860	0.5539
K2	0.9366	0.9487±0.0009	0.5862±0.0022	0.6418±0.0012	0.5344±0.00030	0.5526±0.0033	0.5860±0.0018	0.5539±0.0025
K3	0.9366±0.0035	0.9487±0.0011	0.5862±0.0004	0.6418±0.0015	0.4715±0.00030	0.5526±0.0011	0.586±0.00160	0.5539±0.0028
K4	0.9128±0.0011	0.9435±0.0009	0.7740±0.0035	0.9009±0.0028	0.4715±0.0018	0.5885±0.0004	0.5860±0.0035	0.7119±0.0033
K5	0.9128±0.0007	0.9435±0.0035	0.7740±0.0022	0.9009±0.0018	0.4715±0.0028	0.5885±0.0033	0.5860±0.0007	0.7119±0.0025
K6	0.9366	0.9435±0.0028	0.7740±0.0033	0.9009±0.0014	0.4715±0.0004	0.5885±0.0016	0.5860±0.0035	0.7119±0.0015
h_8								
K1	0.9366±0.0007	0.9589±0.0016	0.5862±0.0004	0.6344±0.0016	0.5344±0.00030	0.8878±0.0011	0.5860±0.0033	0.5591±0.0004
K2	0.9128±0.0011	0.9435±0.0014	0.7740±0.0012	0.9009±0.00030	0.4715±0.0018	0.5885±0.0022	0.5860±0.0031	0.7119±0.0004
K3	0.9333±0.0023	0.9589±0.0022	0.7816±0.00030	0.8465±0.0019	0.7405±0.0004	0.8878±0.0018	0.5860±0.0007	0.8091±0.0004
K4	0.9366±0.0004	0.9230±0.0004	0.5862±0.0018	0.5386±0.0011	0.5344±0.0009	0.4633±0.0033	0.5860±0.0033	0.4981±0.0025
K5	0.9366	0.9333	0.5862	0.6344	0.5344	0.4692	0.5860	0.5394
K6	0.9366±0.0016	0.9333±0.0014	0.5862±0.0018	0.6344±0.0016	0.5344±0.0033	0.4692±0.0012	0.5860±0.0028	0.5394±0.0025
GNB	0.9435		0.8701±0.0019		0.6989±0.0016		0.7965±0.0012	
J48	0.9409±0.0023		0.8104±0.0015		0.6345±0.0011		0.7965±0.0018	
SVM (RBF)	0.9100±0.0009		0.4765±0.0012		0.6211±0.0030		0.5393±0.0031	
SVM (ploynomial)	0.9400±0.00030		0.5101±0.0033		0.6327±0.0019		0.5648±0.0028	
SVM (linear)	0.9333±0.0004		0.6344±0.0011		0.6692±0.0004		0.5394±0.0025	
SVM (sigmoid)	0.9076±0.0012		0.7558±0.0014		0.4685±0.0033		0.5785±0.0015	

SVM – Support vector machine; GNB – Gaussian NB; TPR – True positive rate; RBF – Radial basis function; PPV – Positive predictive value; AAC – Average accuracy

Table 7: For the first situation, predict diabetes using principal component analysis to minimize dimensionality

	AAC		PPV_M		TPR_M		$F_{1,M}$	
	M1	M2	M1	M2	M1	M2	M1	M2
h_1								
K1	0.9538±0.0012	0.9133±0.0023	0.8643±0.0004	0.5475±0.0019	0.7104±0.0033	0.5868±0.0004	0.7799±0.0011	0.5665±0.0011
K2	0.9667±0.0007	0.9433±0.0015	0.8552±0.0004	0.9278±0.0018	0.7289±0.0016	0.7591±0.0034	0.7870±0.0007	0.8352±0.0022
K3	0.9667±0.0014	0.9433±0.00030	0.8552±0.0009	0.9278±0.0012	0.7289±0.0011	0.7591±0.0011	0.7870±0.0011	0.8352±0.0019
K4	0.9538±0.0009	0.9433±0.0023	0.8643±0.0016	0.9278±0.00030	0.7104±0.0028	0.7591±0.0007	0.7799±0.0016	0.8352±0.0004
K5	0.9538±0.0015	0.9233±0.0018	0.8643±0.0023	0.5397±0.0004	0.7104±0.0033	0.6171±0.0004	0.7799±0.0014	0.5758±0.0018
K6	0.9467±0.0007	0.9300±0.0019	0.9297±0.00030	0.9127±0.0018	0.7894±0.0011	0.6380±0.0015	0.8539±0.0028	0.7510±0.0022
h_2								
K1	0.9567±0.0014	0.9433±0.0018	0.5053±0.0016	0.9278±0.0009	0.6317±0.0033	0.7591±0.0035	0.9073±0.0011	0.8352±0.0004
K2	0.9333±0.0034	0.9567±0.0009	0.9222±0.0023	0.5053±0.0004	0.6682±0.0012	0.6317±0.00030	0.7749±0.0028	0.5620±0.0034
K3	0.9667±0.0022	0.9333±0.0015	0.8552±0.0019	0.922±0.00162	0.7289±0.0011	0.6682±0.0031	0.7870±0.0007	0.7749±0.0004
K4	0.9567±0.0014	0.9433±0.0018	0.5053±0.0028	0.9278±0.00030	0.6317±0.0011	0.7591±0.0016	0.5620±0.0028	0.8352±0.0014
K5	0.9633±0.0023	0.9433±0.0015	0.8467±0.0018	0.9278±0.0004	0.6812±0.0016	0.7591±0.0035	0.7550±0.0007	0.8352±0.0004
K6	0.9533±0.0034	0.9433±0.0033	0.4933±0.0011	0.9278±0.00030	0.6299±0.0018	0.7591±0.0016	0.5533±0.0042	0.8352±0.0034
h_3								
K1	0.9538±0.0012	0.9133±0.0007	0.8643±0.0019	0.5475±0.0011	0.7104±0.0009	0.5868±0.0028	0.7799±0.0018	0.5665±0.0012
K2	0.9667±0.0014	0.9433±0.0015	0.8552±0.0007	0.9278±0.00030	0.7289±0.0004	0.7591±0.0011	0.7870±0.0004	0.8352±0.0018
K3	0.9333±0.0028	0.9133±0.0022	0.9222±0.00030	0.5475±0.0018	0.6682±0.0004	0.5868±0.0035	0.7749±0.00030	0.5665±0.0007
K4	0.9538±0.0031	0.9300±0.0009	0.8643±0.0023	0.9127±0.0004	0.7104±0.0015	0.6380±0.0016	0.7799±0.0028	0.7510±0.0042
K5	0.9300±0.0004	0.9133±0.0033	0.9127±0.0011	0.5475±0.0011	0.6380±0.0018	0.5868±0.00030	0.7510±0.0007	0.8352±0.0028
K6	0.9500±0.0042	0.9266±0.0018	0.4871±0.0028	0.5379±0.0007	0.6280±0.00030	0.6351±0.0028	0.5488±0.0018	0.5825±0.0004
h_4								
K1	0.9567±0.0015	0.9467±0.0019	0.5053±0.0022	0.9297±0.0009	0.6317±0.0011	0.7894±0.0012	0.9073±0.0014	0.8539±0.0012
K2	0.9567±0.0007	0.9589±0.0042	0.9444±0.0019	0.9406±0.0028	0.8510±0.0014	0.7081±0.0004	0.8880±0.0011	0.8952±0.0035
K3	0.9567±0.0023	0.9589±0.0033	0.9444±0.0018	0.9406±0.0016	0.8510±0.00030	0.7081±0.0016	0.8880±0.0028	0.8952±0.0042
K4	0.9567±0.0007	0.9589±0.0035	0.9444±0.0014	0.9406±0.0028	0.8510±0.0023	0.7081±0.0004	0.8880±0.0011	0.8952±0.0035
K5	0.9743±0.0004	0.9300±0.0014	0.9891±0.0042	0.9127±0.00030	0.9327±0.0022	0.6380±0.0018	0.9201±0.0028	0.7510±0.0004
K6	0.9692±0.0033	0.9300±0.0042	0.9831±0.0007	0.9127±0.0011	0.793±0.00280	0.6380±0.0042	0.8880±0.0031	0.7510±0.0023
GNB	0.9589±0.0012		0.9406±0.0004		0.7081		0.8952±0.0012	
J48	0.9538±0.0016		0.9001±0.00030		0.7051±0.0004		0.7907±0.0023	
SVM (RBF)	0.9128±0.0016		0.7740±0.0014		0.4715±0.0007		0.5860±0.0011	
SVM (ploynomial)	0.9500±0.0019		0.8879±0.0033		0.6404±0.0011		0.9100	
SVM (linear)	0.9333±0.0009		0.9222±0.0015		0.6682		0.7749	
SVM (sigmoid)	0.9128±0.0023		0.7740±0.0012		0.4715±0.0018		0.5860	

SVM – Support vector machine; GNB – Gaussian NB; TPR – True positive rate; RBF – Radial basis function; PPV – Positive predictive value; AAC – Average accuracy

Table 8: For the second situation, predict diabetes using principal component analysis to minimize dimensionality

	AAC		PPV_M		TPR_M		$F_{1,M}$	
	M1	M2	M1	M2	M1	M2	M1	M2
h_1								
K1	0.9133±0.0019	0.9667±0.0016	0.5475±0.0004	0.8552±0.0033	0.5868±0.0011	0.7289±0.0004	0.5665±0.0011	0.7870±0.0022
K2	0.9433±0.0034	0.9847±0.0028	0.9278±0.0022	0.9415±0.0018	0.7591±0.0019	0.9638±0.00030	0.8352±0.0007	0.9525±0.0016
K3	0.9433±0.0018	0.9847±0.0009	0.9278±0.0016	0.9415±0.0023	0.7591±0.0014	0.9638±0.0011	0.8352±0.0011	0.9525±0.0004
K4	0.9433±0.0028	0.9794±0.0018	0.9278±0.0035	0.9386±0.0007	0.7591±0.0007	0.9082±0.0035	0.8352±0.0009	0.9231±0.0016
K5	0.9233±0.0015	0.9538±0.0009	0.5397±0.0004	0.8643±0.0014	0.6171±0.00030	0.7104±0.0004	0.5758±0.0022	0.7799±0.0033
K6	0.9467±0.0016	0.9847±0.0035	0.9297±0.0028	0.9415±0.0007	0.7894±0.0007	0.9638±0.0034	0.8539±0.0031	0.9525±0.0026
h_2								
K1	0.9433±0.0018	0.9794±0.0028	0.9278±0.0007	0.9386±0.00030	0.7591±0.0034	0.9082±0.0004	0.8352±0.0018	0.9231±0.0016
K2	0.9333±0.00030	0.9743±0.0014	0.9222±0.0033	0.9859±0.0018	0.6682±0.0023	0.8484±0.0028	0.7749±0.0011	0.9120±0.0035
K3	0.9333±0.0004	0.9743±0.0011	0.9222±0.0012	0.9859±0.0009	0.6682±0.0016	0.8484±0.0015	0.7749±0.0019	0.9120±0.0011
K4	0.9567±0.0014	0.9743±0.0004	0.5053±0.0018	0.9859±0.00030	0.6317±0.0004	0.8484±0.0019	0.9073±0.0022	0.9120±0.0011
K5	0.9433±0.0033	0.9743±0.0014	0.9278±0.0015	0.9859±0.0023	0.7591±0.00030	0.8484±0.0011	0.8352±0.0018	0.9120±0.0035
K6	0.9533±0.0011	0.9800±0.0015	0.4933±0.0033	0.9176±0.0014	0.6299±0.0018	0.9176±0.0016	0.5533±0.0011	0.9176±0.0042
h_3								
K1	0.9133±0.0004	0.9538±0.0014	0.5475±0.0018	0.8643±0.0011	0.5868±0.0011	0.7104±0.0016	0.5665±0.0028	0.7799±0.0022
K2	0.9767±0.00030	0.9743±0.0009	0.8991±0.0004	0.9859±0.0023	0.9157±0.0014	0.8484±0.0011	0.9073±0.0035	0.9120±0.0016
K3	0.9567±0.0034	0.9743±0.0004	0.5053±0.0018	0.9859±0.0011	0.6317±0.0018	0.8484±0.0007	0.5620±0.0004	0.9120±0.0028
K4	0.9300±0.0014	0.9743±0.0015	0.9127±0.0033	0.9859±0.0019	0.6380±0.0011	0.8484±0.0018	0.7510±0.0011	0.9120±0.0028
K5	0.9300±0.0011	0.9743±0.0004	0.9127±0.0022	0.9859±0.0011	0.6380±0.0018	0.8484±0.0004	0.7510±0.0030	0.9120±0.0035
K6	0.9633±0.0015	0.9800±0.0004	0.8467±0.0023	0.9176±0.0018	0.6812±0.0011	0.9176±0.0028	0.7550±0.0015	0.9176±0.0018
h_4								
K1	0.9467±0.0028	0.9846±0.0014	0.9297±0.0009	0.9458±0.0016	0.7894±0.0007	0.9386±0.0004	0.8539±0.0033	0.9421±0.0035
K2	0.9633±0.0004	0.9867±0.0018	0.8462±0.0011	0.9642±0.0011	0.7251±0.0019	0.9212±0.0023	0.7810±0.0009	0.9422±0.0018
K3	0.9641±0.0004	0.9867±0.0018	0.9168±0.0004	0.9642±0.0028	0.7689±0.00030	0.9212±0.0014	0.8364±0.0033	0.9422±0.0028
K4	0.9641±0.0033	0.9846±0.0015	0.9168±0.0023	0.9458±0.0012	0.7689±0.0018	0.9386±0.0019	0.8364±0.0028	0.9421±0.0042
K5	0.9433±0.0011	0.9767±0.0014	0.9278±0.0022	0.8991±0.0011	0.7591±0.0015	0.9157±0.0004	0.8352±0.0035	0.9073±0.0009
K6	0.95330.0023	0.9692±0.0004	0.4933±0.0009	0.9831±0.0028	0.6299±0.0034	0.7930±0.0022	0.5533±0.0016	0.8880±0.0016
GNB	0.9641±0.0042		0.9406		0.76690.0023		0.8038±0.0012	
J48	0.9589±±0.0012		0.9309±0.0011		0.7081±0.0004		0.8952±0.0011	
SVM (RBF)	0.9333±0.0009		0.9222±0.0014		0.6682±0.0033		0.7749±0.0016	
SVM (ploynomial)	0.9538±0.0004		0.9001±0.0015		0.7051±0.00030		0.7907±0.0022	
SVM (linear)	0.9435±0.0023		0.8847±0.0033		0.6989±0.0012		0.8387±0.0011	
SVM (sigmoid)	0.9333±0.0019		0.9222±0.0016		0.6682		0.7749±0.0014	

SVM – Support vector machine; GNB – Gaussian NB; TPR – True positive rate; RBF – Radial basis function; PPV – Positive predictive value; AAC – Average accuracy

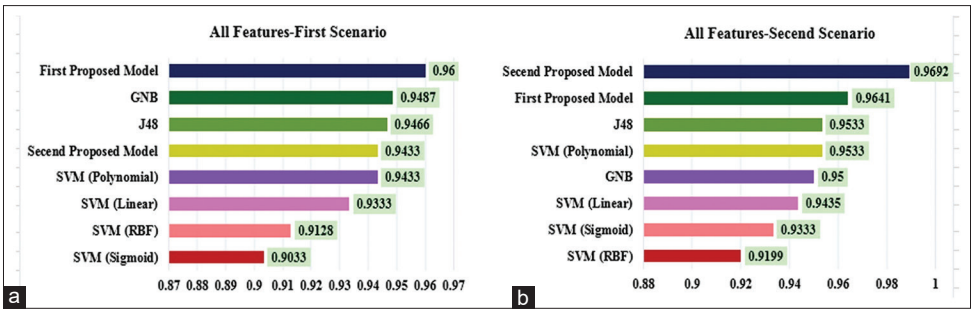

Figure 8: Diabetes can be predicted using all of these features (13). (a). In the first scenario, that is, the presence of an outlier, M1 had a greater AAC in predicting diabetes than did the other models. (b). In the second scenario, after outlier rejection, M2 had a greater AAC for predicting diabetes than did the other models

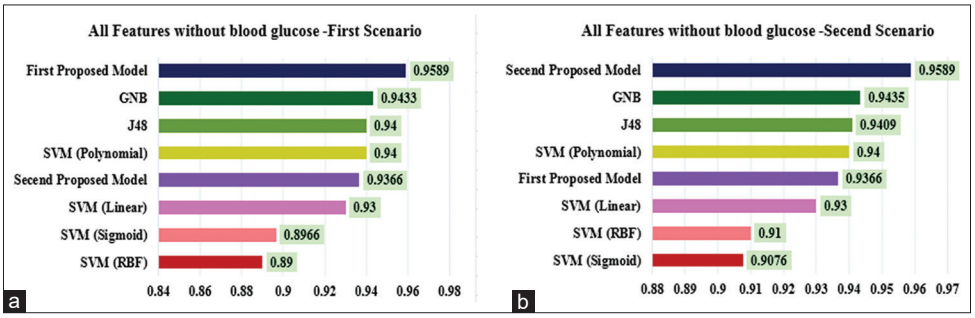

Figure 9: Diabetes can be predicted using all features without blood glucose (12). (a). In the first scenario, that is, the presence of an outlier, M1 had a greater AAC in predicting diabetes than did the other models. (b). In the second scenario, after outlier rejection, M2 had a greater AAC in predicting diabetes than did the other models, and compared to the other experiments, in this case, the AAC decreased

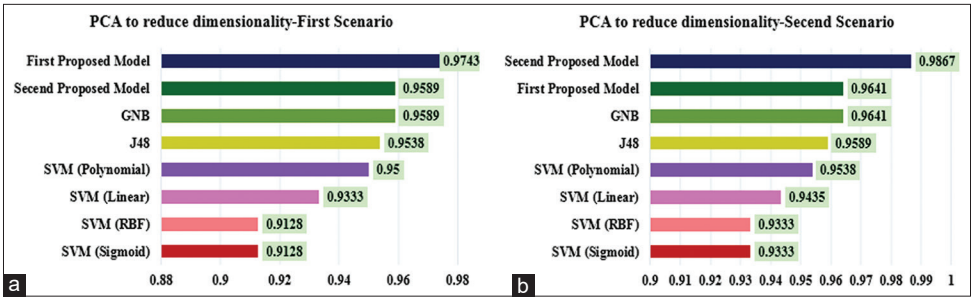

Figure 10: Diabetes can be predicted by using principal component analysis to reduce dimensionality. Compared to other experiments, in this case, the highest AAC value is obtained. (a) In the first scenario, that is, the presence of an outlier, M1 had a greater AAC in predicting diabetes than did the other models. (b). In the second scenario, after outlier rejection, M2 had a greater AAC in predicting diabetes than did the other models, and compared to the other experiments, in this case, the AAC decreased

similar performances and are better than the first model, which takes the presence of outliers into account.

In addition, by comparing the results obtained in Figure 9, we conclude that when predicting diabetes in both scenarios without blood glucose, the performance is lower than that when using all features to predict diabetes.

From Figure 10, we can find the appropriate PCA for the dataset. Using all the features did not have good performance; however, when we used the PCA, in both scenarios, the two proposed models had the optimal performance compared to the other models.

In addition the number of clusters in each class was chosen from 2, 3…,15 in both clustering algorithms employed in this study. To identify the number of ideal clusters, we use a greedy search technique. To be more exact, we set the number of clusters for all classes to 1 from the start. Then, for the class labeled 0, we obtain the optimal number of clusters. Then, there is number 1, etc. Once the number of clusters of a class is received, it will be fixed until we determine all other classes' cluster numbers. We also experimented several times and concluded that to obtain suitable performance in the two proposed models in both scenarios, the number of clusters in the first class affects the performance of the models, and the change in the number of second- and third-class clusters has less effect on one, as shown in Figure 11. When the number of first-class clusters reaches 3, the performance of the proposed models is more appropriate.

In contrast, the number of clusters in the other two classes is 1 or 2. This is because, in the dataset, the number of

samples from the first class is much greater than that from the other two classes. For comparison with the methods in other papers, we used fivefold cross-validation in all the experiments. To balance the data for each class, we assigned the following weights to each class: 1.18 for class C0, 9.71 for class C1, and 18.87 for the C2 class.

Finally, the two proposed hybrid models performed better than the single models investigated on the IPDD dataset in terms of performance measures. According to Table 9 and Figure 12, comparing the proposed method with the method presented by Soukaena Hassan *et al*.[14] shows that the proposed model outperforms the other two models both in terms of presence and after outlier rejection

Discussion and Future Work

The primary objective of this study was to develop and assess the efficacy of two novel machine-learning hybrid models for predicting type 2 diabetes mellitus (DM). The first model combines the K-medoids clustering algorithm with a GNB classifier, using KDE to enhance classification accuracy. The second model uses the K-means clustering algorithm with the same classifier. To reduce dimensionality, we applied PCA, and we evaluated the model performance through rigorous fivefold cross-validation.

Type 2 DM is a pervasive chronic metabolic condition that affects millions of people worldwide. Timely diagnosis and intervention are paramount for mitigating complications and improving patient prognosis. Machine-learning methodologies offer a promising avenue for predictive analytics in type 2 DM, prompting the development and evaluation of these innovative hybrid models.

Recent research has underscored the utility of diverse machine-learning frameworks in this domain, yielding varying insights and outcomes. The robustness of the proposed models in handling outliers emerges as a notable advantage. Leveraging K-medoids and K-means clustering algorithms, these models demonstrate resilience to outlier influence. Moreover, the GNB classifier, grounded in KDE, exhibits commendable efficacy in probabilistic feature classification. PCA has emerged as a pivotal tool for dimensionality reduction, preserving essential data facets while streamlining computational complexity.

Despite the promising results, this study is not without limitations. Firstly, the models were tested on a relatively small dataset, which might not capture the full spectrum of variability present in larger and more diverse populations. Secondly, while PCA effectively reduced dimensionality, it also potentially discarded some minor yet informative features that could contribute to the prediction accuracy. Thirdly, the computational complexity of the hybrid models, especially with large datasets, presents scalability concerns that need to be addressed. Lastly, the reliance on specific clustering algorithms (K-medoids and K-means) and the GNB classifier may limit the generalizability of the models to datasets with different distributions or underlying characteristics.

Future research should explore several avenues to address these limitations and enhance the models' applicability. One critical area is validating these models on larger and

Table 9: Comparison of diabetes prediction models in terms of AAC performance criteria

Researchers	Proposed model	Presence of the outlier (%)	After the outlier rejection (%)
Soukaena Hassan *et al*.[14]	Designing a diabetes hybrid diagnosis system by combining KNN and ID3 algorithms	75.50	98.25
Current study	Combination of clustering and classification method	97.43	98.67

AAC – Average accuracy

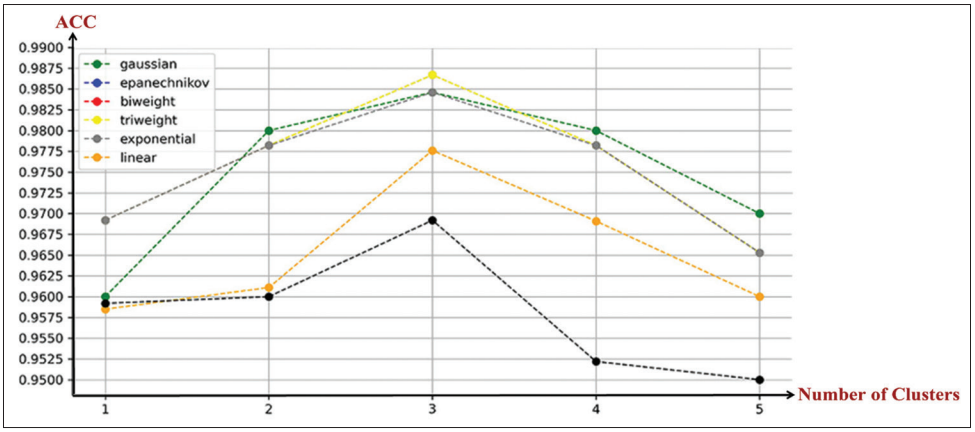

Figure 11: Changes in the number of first-class clusters can have an impact on the performance of the two models in different scenarios. The number of first- and second-class clusters has less impact than the number of first-class clusters

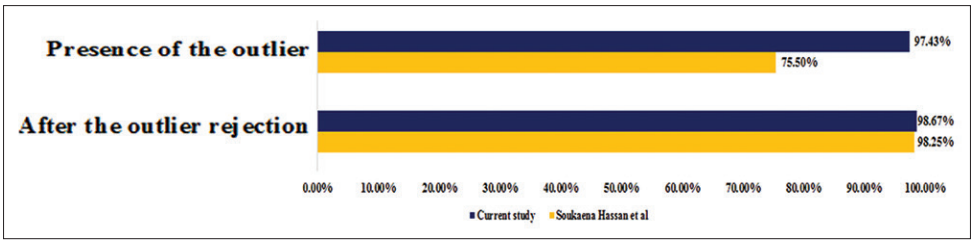

Figure 12: Comparison of diabetes prediction models in terms of AAC performance criteria

more heterogeneous datasets to ensure their robustness and generalizability. In addition, investigating alternative dimensionality reduction techniques that retain more informative features without significantly increasing computational complexity could improve model performance.

Exploring different clustering algorithms and classifiers may yield better results for specific datasets. For instance, hierarchical clustering or density-based clustering methods might offer advantages in handling nonlinear data structures. Similarly, employing ensemble learning techniques or deep learning approaches can enhance predictive accuracy and adaptability.

Moreover, incorporating additional patient data such as genetic information, lifestyle factors, and real-time health monitoring data could provide a more comprehensive model for predicting type 2 DM. Finally, developing methods to dynamically update and refine the models as new data becomes available would ensure their continued relevance and accuracy.

Conclusion

This study presents two hybrid models for type 2 diabetes prediction, each addressing distinct scenarios. In the first scenario (M1), a GNB based on KDE was used with K-medoids clustering to group data with statistical similarity and insensitivity to outliers. In the second scenario (M2), outliers were removed before applying K-means clustering to group the remaining data. A GNB classifier based on KDE was then used for diabetes classification. The dataset used in this study included 1000 physical examination samples, including outliers, from the Medical City Hospital's laboratory. The proposed models were evaluated using classification metrics such as accuracy, precision, F1-score, and recall and compared to other algorithms, including GNB with different kernel functions, support SVM with different kernel functions, and DT. The experimental results showed that the proposed hybrid models performed better across multiple evaluation metrics, outperforming other algorithms like GNB, DT (J48), and SVM with polynomial and sigmoid kernels.

Availability of data and materials

The data used were from a publicly available dataset[42] (https://data.mendeley.com/datasets/wj9rwkp9c2/1) (Note: Of course, it should be noted that the dataset in this link does not have the attribute value FBS. Through correspondence with the person responsible for this dataset,[28] we obtained the values of this feature and added them to the dataset.).

Acknowledgments

We would like to express our profound gratitude to Dr. Silva Hovsepian for her essential advice and knowledge on the medical aspects of this work. She is an assistant professor at the Metabolic Liver Diseases Research Center at Isfahan University of Medical Sciences. We are sincerely appreciative of her help, as her contributions had a significant role in determining the course of our research. I appreciate Dr. Hovsepian.

Financial support and sponsorship

Nil.

Conflicts of interest

There are no conflicts of interest.

References

1. Hezagirwa B, Riewpaiboon A, Chanjaruporn F. Exploring cost drivers to improve disease management: The case of type 2 diabetes at a tertiary hospital in Burundi, Africa. J Public Health Afr 2023;14:2266.
2. Żuchnik M, Rybkowska A, Szczuraszek P, Szczuraszek H, Bętkowska P, Radulski J, *et al*. Type 2 diabetes-factors of occurrence and its complications. Qual Sport 2023;10:32-40.
3. Beljić ZT. Prediabetes: From diagnosis to prognosis. Galenika Med J 2022;1:57-61.
4. Wiesmann UN, DiDonato S, Herschkowitz NN. Effect of chloroquine on cultured fibroblasts: Release of lysosomal hydrolases and inhibition of their uptake. Biochem Biophys Res Commun 1975;66:1338-43.
5. Kant R, Davis A, Verma V. Maturity-onset diabetes of the young: Rapid evidence review. Am Fam Physician 2022;105:162-7.
6. Warth J, Desforges JF. Determinants of intracellular pH in the erythrocyte. Br J Haematol 1975;29:369-72.
7. IDF Diabetes Atlas. Available from: https://diabetesatlas.org/. [Last accessed on 2024 Feb 15].
8. WHO. Diabetes. Available from: https://www.who.int/health-topics/diabetes#tab=tab_1. [Last accessed on 2024 Feb 15].
9. Sun X, Qourbani A. Combining ensemble classification and integrated filter-evolutionary search for breast cancer diagnosis. J Cancer Res Clin Oncol 2023;149:10753-69.
10. Farnoosh R, Abnoosian K. A robust innovative pipeline-based machine learning framework for predicting COVID-19 in Mexican patients. Int J Syst Assur Eng Manage 2024;15:3466-

84. doi: 10.1007/s13198-024-02354-3.
11. Kumar P, Ganesh S, Vijay AA, Devaraj D. A hybrid colony fuzzy system for analyzing diabetes microarray data. In 2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). IEEE; 2013. p. 104-11.
12. Abnoosian K, Farnoosh R, Behzadi MH. A pipeline-based framework for early prediction of diabetes. J Health Biomed Inform 2023;10:125-40.
13. Abnoosian K, Farnoosh R, Behzadi MH. Prediction of diabetes disease using an ensemble of machine learning multi-classifier models. BMC Bioinformatics 2023;24:337.
14. Hassan S, Karbat AR, Zaki S. Towfik. Propose Hybrid KNN-ID3 for Diabetes Diagnosis System. Intern J Sci Engineer Res 2014;5:1087-104.
14. Yamamoto JM, Pylypjuk C, Sellers E, McLeod L, Wicklow B, Sirski M, *et al.* Maternal and neonatal outcomes in pregnancies with type 2 diabetes in first nation and other manitoban people: A population-based study. CMAJ Open 2022;10:E930-6.
15. Reinehr, T, Wabitsch M. Type 2 diabetes mellitus in children and adolescents. Prevention of type 2 diabetes; 2005. p. 21-40.
16. Butt MD, Ong SC, Wahab MU, Rasool MF, Saleem F, Hashmi A, *et al.* Cost of Illness analysis of type 2 diabetes mellitus: The findings from a lower-middle income country. Int J Environ Res Public Health 2022;19:12611.
17. Gülümsek E, Keşkek ŞÖ. Direct medical cost of nephropathy in patients with type 2 diabetes. Int Urol Nephrol 2022;54:1383-9.
18. N.D.P. Program. "How Type 2 Diabetes Affects Your Workforce. Available from: https://www.cdc.gov/diabetes/prevention/how-type2-affects-workforce.htm.
19. Zhou H, Xin Y, Li S. A diabetes prediction model based on boruta feature selection and ensemble learning. BMC Bioinformatics 2023;24:224.
20. Mehrpour O, Saeedi F, Abdollahi J, Amirabadizadeh A, Goss F. The value of machine learning for prognosis prediction of diphenhydramine exposure: National analysis of 50,000 patients in the United States. J Res Med Sci 2023;28:49.
21. Lenatti M, Carlevaro A, Guergachi A, Keshavjee K, Mongelli M, Paglialonga A. A novel method to derive personalized minimum viable recommendations for type 2 diabetes prevention based on counterfactual explanations. PLoS One 2022;17:e0272825.
22. Gohari K, Kazemnejad A, Mohammadi M, Eskandari F, Saberi S, Esmaieli M, *et al*. A bayesian latent class extension of naive bayesian classifier and its application to the classification of gastric cancer patients. BMC Med Res Methodol 2023;23:190.
23. Mostafa SA, Elzanfaly DS, Yakoub AE. A machine learning ensemble classifier for prediction of brain strokes. Int J Adv Comput Sci Appl 2022;13.
24. Sisodia D, Sisodia DS. Prediction of diabetes using classification algorithms. Procedia Comput Sci 2018;132:1578-85.
25. Chaudhary AM, Sanaullah A, Hanif M, Almazah MM, Albasheir NA, Al-Duais FS. Efficient monitoring of a parameter of non-normal process using a robust efficient control chart: A comparative study. Mathematics 2023;11:4157.
26. Maimon O, Rokach L. (Eds.). Data mining and knowledge discovery handbook. New York: Springer; Vol. 2. 2005.
27. Dalal MA, Harale ND, Kulkarni UL. An iterative improved k-means clustering. In Intern Conf Advances in Computer Engineering 2011. p. 25-8.
28. Syarif I, Prugel-Bennett A, Wills G. Unsupervised clustering approach for network anomaly detection. In Networked Digital Technologies: 4th International Conference, NDT 2012, Dubai, UAE, April 24-26, 2012. Proceedings, Part I Springer Berlin Heidelberg. 2012;4:135-45.
29. An Improved k-Medoids Clustering Algorithm. In: The 2nd International Conference on Computer and Automation Engineering (ICCAE); 2010. p. 132-5.
30. Velmurugan T, Santhanam T. Computational complexity between K-means and K-medoids clustering algorithms for normal and uniform distributions of data points. J Comput Sci 2010;6:363.
31. Fadhil ZM. Hybrid of K-means clustering and naive Bayes classifier for predicting performance of an employee. Periodic Engineer Natural Sciencesissn 2021;9:799-807.
32. Husejinovic A. Credit card fraud detection using naive Bayesian and c4. 5 decision tree classifiers. 2020;4:1-5.
33. Hand DJ, Yu KJ. Idiot's bayes-not so stupid after all? Intern Statistical Rev 2001;69:385-98.
34. Solorio-Ramírez JL, Saldana-Perez M, Lytras MD, Moreno-Ibarra MA, Yáñez-Márquez C. Brain hemorrhage classification in CT scan images using minimalist machine learning. Diagnostics (Basel) 2021;11:1449.
35. Huang S, Cai N, Pacheco PP, Narrandes S, Wang Y, Xu W. Applications of Support vector machine (SVM) learning in cancer genomics. Cancer Genomics Proteomics 2018;15:41-51.
36. Janicka M, Lango M, Stefanowski J. Using information on class interrelations to improve classification of multiclass imbalanced data: A new resampling algorithm. Int J Appl Math Comput Sci 2019;29.
37. Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognit 2011;44:1761-76.
38. Quinlan JR. Induction of decision trees. Mach Learn 1986;1:81-106.
39. Song YY, Lu Y. Decision tree methods: Applications for classification and prediction. Shanghai Arch Psychiatry 2015;27:130-5.
40. Parzen E. On estimation of a probability density function and mode. Ann Math Stat 1962;33:1065-76.
41. Ahlam R. "Diabetes Dataset", Mendeley Data, V1, 2020. doi: 10.17632/wj9rwkp9c2.1.
42. Cousineau D, Chartier S. Outliers detection and treatment: a review. Intern J Psychol Res 2010;3:58-67.
43. Outlier Detection: Applications and Techniques in Data Mining. In: 2016 6th International Conference – Cloud System and Big Data Engineering (Confluence); 2016. p. 373-7.
44. Patro S. Normalization: A preprocessing stage. 2015. arXiv preprint arXiv:1503.06462.
45. Zhang W, Zhang Z, Chao HC, Tseng FH. Kernel mixture model for probability density estimation in Bayesian classifiers. Data Min Knowl Discov 2018;32:675-707.
46. Haijin JI, Huang S, Xuewei LV, Yaning Wu, Feng Y. Empirical studies of a kernel density estimation based naive bayes method for software defect prediction. IEICE TRANSACTIONS on Information and Systems 2019;102: 75-84.
47. Jackson DA. Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches. Ecology 1993;74:2204-14.
48. Hossin M, Sulaiman MN. A review on evaluation metrics for data classification evaluations. Int J Data Min Knowl Manage Process 2015;5:1.
49. Grandini M, Bagli E, Visani G. Metrics for multi-class classification: an overview. arXiv preprint arXiv:2008.05756. 2020.
50. Bayes, T. LII. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in a letter to John Canton, AMFR S. Philosophical transactions of the Royal Society of London 1763;53:370-418.

Appendix

In this section, of the article, which serves as an appendix for diabetes disease prediction, we delve into a range of key components. This encompasses various types of clustering algorithms (A), classification algorithms (B), data preprocessing (C), kernel functions (D), pseudocode of the proposed model (E), principal component analysis (F), and evaluation metrics (G). These fundamental insights aim to enhance comprehension and facilitate a more nuanced analysis of the primary subject matter, ultimately aiding in the provision of more effective solutions for diabetes disease prediction.

A. Clustering Algorithms

- **K-means Clustering**

The K-means clustering algorithm can be summarized by the following pseudocode, which outlines the steps involved in the process:

1. Select the k value (k is the total number of clusters).
2. Arbitrary Select k initial centers (centroid) c_1, c_2, ..., c_k.
3. For each $1 \le i \le k$, set the cluster C_i to be the set of points in X that are closer to c_i than to any c_j with $j \neq i$.
4. For each $1 \le i \le k$, set $c_i = \frac{1}{|C_i|}\sum_{x \in C_i} x$, i.e., the center of mass of the points in C_i.
5. Repeat Steps 1 and 2 until the clusters C_i and the center's c_i do not change anymore. The partition of X is the set of clusters C_1, C_2, ..., C_k.

- K-medoids Clustering

The K-medoids clustering algorithm can be summarized by the following pseudocode, which outlines the steps involved in the process:

1. Select the k value (k is the total number of clusters).
2. Arbitrary Select k initial centers (medoids).
3. Assign each point to the cluster with the nearest medoid.
4. Calculate the total distance between the object and its cluster medoid.
5. Swap the medoid with a nonmedoid point.
6. Recalculate the positions of the k medoids.
7. Repeat Steps 1 to 4 until the medoids become fixed.

B. Classification Algorithms

- **Naive Bayes Classifier**

According to [61], $X = \{x_1, x_2, ..., x_N\}$ be training samples, such as the i^{th} feature vector $X_i = [x_{i1}, x_{i2}, ..., x_{id}]$, and let C be a set containing class labels (in this study, we have three classes). This classification method is as follows: For a new test instance $X = [x_1, x_2, ..., x_d]$, for all classes, the $c \in C$ classifier returns the $\hat{c}$ class with the highest class probability or class conditional probability.

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \; P(c|x) \tag{1}$$

This idea of Bayesian inference has been known since Bayesian time. Now, using the Bayesian rule, the conditional probability (likelihood) is defined as follows:

$$P(c|X) = \frac{P(X|c)P(c)}{P(X)} \tag{2}$$

By substituting Eq. 4 into Eq. 3, we obtain the following equation:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \, P(c|x) = \underset{c \in C}{\operatorname{argmax}} \, \frac{P(X|c)P(c)}{P(X)} \tag{3}$$

Because P(X) does not change for each class and our goal is to find the most likely class with the same class for each class, we can drop it in $\frac{P(X|c) \mid P(c)}{P(c)}$ to the denominator, so we can select the most likely class with the simplified equation below:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\ P(c|x) = \underset{c \in C}{\operatorname{argmax}}\, P(X|c)P(c) \tag{4}$$

Now, by placing the test sample vector X = [x_1, x_2, ..., x_d] in Eq. 6, we obtain the following equation:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\ P(c|x_1, x_2, \ldots, x_d) = \underset{c \in C}{\operatorname{argmax}}\ P(x_1, x_2, \ldots, x_d|c)P(c) \tag{5}$$

where P(x_1, x_2, ..., x_d|c) is the likelihood and P(c) is the prior probability and is obtained as follows:

$$P(c) = \frac{N_c}{N} \tag{6}$$

where N_c is the total number of training samples from class c ($c \in C$) and is N the total number of training samples. Now, according to the NB conditional independence assumption, which expresses the independence of probabilities P(x_i|c) from a given class c, "naively" P(x_1, x_2, ..., x_d|c) can be multiplied as follows:

$$P(x_1, x_2, \ldots, x_d|c) = P(x_1|c) \times P(x_2|c) \times \ldots \times P(x_d|c) \tag{7}$$

Ultimately, by placing Eq. 9 in Eq. 7, the final classifier for selecting the most likely class by the NB is converted as follows:

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c) \prod_{i=1}^{d} P(x_i|c) \tag{8}$$

- **J48 Classifier Algorithm:**

This algorithm can be summarized by the following pseudocode, which outlines the steps involved in the process
1. An attribute for the root node is selected.

1. Generate a branch for each possible attribute value.
2. The sample is divided into multiple subsets, each subset of which corresponds to a branch of the root node.
3. The process was repeated in reverse for each branch until all samples had the same classification.

In the DT (J48) classifier, nodes are determined by the information gain criterion. The J48 classifier based on Eq. 11 calculates the value of the information gain for each attribute in each iteration and chooses the attribute with the highest value of the information gain as the node for the current iteration. The information gain G(X, A) defines the attribute A on the set-in terms of the set X of available samples as follows:

$$G(X, A) = H(X) - \sum_{v \in Values(A)} \frac{|X_v|}{X} H(X_v) \tag{9}$$

where G denotes the information gain, H is the entropy, Values (A) is the category of all possible values for attribute A, and X_v is the number of subsamples of X that have the value of v for attribute A, i.e., $X_v = \{x \in X | A(x) = v\}$.

- **Kernel Density Classifier:**

If X_1, X_2, ..., X_n are samples of a continuous unknown probability density function f, then the probability density function $\hat{f} : \mathbb{R} \to \mathbb{R}$ estimated by KDE is defined as follows:

$$\hat{f}_H(X) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{X - X_i}{h}\right) \tag{10}$$

Here, h > 0 is the smoothing parameter of the KDE estimator, known as the bandwidth, n is the number of samples, and $K(.) : \mathbb{R} \to \mathbb{R}$ is the nonnegative kernel function with the following constraint.

$$\int_{-\infty}^{+\infty} K(x)\,dx = 1 \tag{11}$$

C. Data Preprocessing

- **Outlier Rejections**

We used the following mathematical equation to reject outliers:

$$OR(x)=\begin{cases} x. & \text{if } Q_1-\frac{3}{2}\times IQR \le x \le Q_1+\frac{3}{2}\times IQR \\ \text{reject.} & \text{otherwise} \end{cases} \tag{12}$$

where x is the number of occurrences of a feature vector in n dimensions; $x \in \mathbb{R}^n$, the first quartile, third quartile, and interquartile range of the qualities are represented by Q_1, Q_3, and IQR, respectively, where Q_1, Q_3, and $IQR \in \mathbb{R}^n$.

- **Min–Max normalization:**

We used the following mathematical equation for normalization:

$$\acute{x}=\frac{x-\min}{\max-\min} \tag{13}$$

where x and $\acute{x}$ are equal to the original data and the converted value, respectively.

D. Kernel Functions

In this study, we utilized 8 different types of kernel functions, which are listed in Table 1.

Table 1: Kernels function

Kernels	Equations
Gaussian	$K(u)\propto\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{u^2}{2}\right)$
Epanechnikov	$K(u)=\frac{3}{4}(1-u^2)I(\lvert u\rvert\le 1)$
Biweight (quartic)	$K(u)=\frac{15}{16}(1-u^2)I(\lvert u\rvert\le 1)$
Triweight	$K(u)=\frac{35}{32}(1-u^2)I(\lvert u\rvert\le 1)$
Exponential	$K(u)\propto\frac{1}{\sqrt{2\pi}}\exp(-u)$
Linear	$K(u)\propto 1-U \text{ if } u<0$

E. Pseudo code of the proposed model

Input: Training data sample $\{(c_1,X_1), \dots, (c_C,X_n)\}$, number of HSC $[n_1, \dots, n_c]$, test data X.

Output: Classification result c_d.

1. For (each $c_i \subseteq C$) {
2: Compute $\{(c_i^1.X_1^{n_i}).....(c_i^{n_i}.X_i^{n_i})\}$ by using K-means or K-medoids algorithms;
3. For each $c_i^1 \in c_i$ {
4. Compute prior probability using Eq.(2) from section (2-4).
5. Compute bandwidth $h_{i.j}^k=\text{dia.}\left[h_{i.j}^k,\dots,h_{i.r}^k\right]$ by using Eqs. (4) or (5), and (6) from section (2-4).
6. For each $x_j \in X$
7. Compute likelihood $P(x_j|c_i^k)$ by using Eq. (3) with $h_{i.j}^k$ from section (2-4).
8. }
9. }
10. Compute posterior probability $P(c_i|X)$ by using (2) from section (2-4).
11: end for each
12: Choose the maximum posterior probability $\hat{c}=\text{argmax}_{1\le c\le C}\{p(c_1|X).\cdots.p(c_k\mid X)\}$

F. Mathematical representation of the PCA factor extraction model

The mathematical representation from the PCA model can be expressed as follows:

$$\text{Factor}_i = T_{i1}X_1 + T_{i2}X_2 + T_{ik}X_k \quad (i = 1, 2, \ldots, m) \tag{14}$$

PCAFactor_i represents the i principal component factor; T_{ij} represents the i principal component factor's load on the j index; m represents the number of principal component factors; and k represents the number of indicators.

G. Evaluation Metrics

In this study, we used various metrics for evaluating the multiclass classification model to measure the effectiveness of the classification, the formulas of which are as follows:

- **Average Accuracy: accuracy (AAC),** The average per-class effectiveness of the classifier

$$\text{ACC} = \frac{\sum_{i=1}^{k} \frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i}}{k} \tag{15}$$

- **Precision_M:** Precision or positive predictive value (PPV), average per-class agreement of the true class labels with those of the classifier

$$\text{PPV}_M = \frac{\sum_{i=1}^{k} \frac{tp_i}{(tp_i + fp_i)}}{k} \tag{16}$$

- **Recall_M:** Sensitivity, recall, hit rate, or true positive rate (TPR), average per-class effectiveness of a classifier to identify class labels

$$\text{TPR}_M = \frac{\sum_{i=1}^{k} \frac{tp_i}{(tp_i + fn_i)}}{k} \tag{17}$$

- **F1 - score_M:** The harmonic mean of the microaverage precision and recall

$$F_{1,M} = \frac{2 \times \text{Precision}_M \times \text{Recall}_M}{\text{Precision}_M + \text{Recall}_M} \tag{18}$$

where the total number of classes is k, the number of microaverages is M, the number of true positives TP represents the number of samples that are predicted to be positive and true, the number of samples that are predicted to be positive and false (FP), the number of samples that are predicted to be negative and true (TN), and the number of samples that are predicted to be negative and false (FN).