Cureus Journal of Computer Science

Open Access Original Article

# Performance Evaluation of Diabetes Prediction Model Based on Imbalanced Dataset Using Feature Selection and Hyperparameter Tuning of Classifiers

Subhash C. Gupta [1] , Noopur Goel [1]

1. Department of Computer Applications, Veer Bahadur Singh Purvanchal University, Jaunpur, IND

**Corresponding authors:** Subhash C. Gupta, csubhashgupta@gmail.com, Noopur Goel, noopurt11@gmail.com

**Received** 04/18/2025
**Review began** 05/23/2025
**Review ended** 08/01/2025
**Published** 08/03/2025



**DOI:**
https://doi.org/10.7759/s44389-025-04842-4

## Abstract

The performance of a prediction model heavily depends on the distribution of class labels in the dataset. An imbalanced dataset always negatively impacts the performance of the prediction model and necessitates the use of appropriate metrics for evaluation. In this paper, research has been conducted on an imbalanced dataset - PIMA Diabetes Dataset, and the Matthews correlation coefficient has been considered as the evaluation metric for the model. Feature selection methods have also been applied to select only relevant features from the dataset to enhance the performance of the metrics. The features selected by different feature selection methods may or may not be the same; selecting only those features present in all selected lists may reduce the size of the processed dataset to a critical level and potentially decrease the performance of the model.

In this paper, a method called weighted index () is proposed, which uses an acceptability parameter (threshold value) to select the most relevant features from feature lists when these lists differ. Furthermore, two additional datasets were created: one in which all missing values were replaced with the mean of the corresponding column, and a second in which samples containing missing values were removed. Feature selection methods were applied to all three datasets, and the weighted index was used to identify common features based on a predefined threshold. Finally, prediction models were built using K-nearest neighbors, decision tree, random forest, and support vector machine, with the Matthews correlation coefficient as the primary evaluation metric to select the best model and analyze its performance.

**Categories:** Data Mining, Data Science Methodologies, Machine Learning (ML)
**Keywords:** diabetes mellitus, imbalanced dataset, mcc, weigthed_index (), knn, svm, random forest, decision tree

## Introduction

Diabetes is a non-communicable disease caused by an increased amount of blood sugar in the body and sometimes by the malfunctioning of β-cells in the pancreas. Diabetes can also lead to other diseases related to the kidneys, nervous system, eyes, and more. Lack of exercise and an unhealthy lifestyle are major reasons behind the current situation. There are three types of diabetes: Type 1 diabetes, also known as juvenile diabetes, is mostly found in children; Type 2 diabetes is found mainly in middle-aged and older adults; and the third type, gestational diabetes, occurs in pregnant women and usually disappears after delivery [1,2]. Once a person develops diabetes, it cannot be cured, but it can be managed through regular physical activity, exercise, medication, and a healthy diet [3].

Today, diabetes has become one of the major threats among the largest global health problems for the world population. According to the report IDF Diabetes Atlas 2025, published by the International Diabetes Federation [1], more than half a billion people (approximately 589 million) worldwide are living with diabetes. A major concern is that this is not yet the peak of its prevalence - millions of new diabetic patients are added each year. The report states that in 2024, approximately 588.7 million people were suffering from diabetes, and the number is growing so rapidly that it is projected to reach 643 million by 2030 and 852.5 million by 2050 [1,2]. India is the second-largest country affected by diabetes, accounting for one out of every six diabetic individuals globally. It was once commonly said that diabetes is a disease of the rich, but today it is found across all income groups and age groups, spreading rapidly [4]. According to the WHO 2016 report, the number of people with diabetes in India increased from 69.2 million to 72.9 million in just two years, from 2015 to 2017 [2].

In the medical profession, diabetes mellitus is one of the most severe problems, requiring the intervention of new technologies and research for proper diagnosis (classification) and care of diabetic patients. Therefore, the classification or prediction of diabetes is a popular research topic among researchers using techniques such as data mining and machine learning. The primary objective of their research is to classify

data as diabetic or non-diabetic and to improve the accuracy of prediction models.

Machine Learning, a branch of Artificial Intelligence, involves the development of techniques that enable computers to learn from past experiences and make decisions. The learning ability in computers refers to their capacity to understand and identify patterns in input data in order to make decisions and perform prediction/classification tasks. Machine learning is performed in two stages: the first stage is the learning stage, which involves building a prediction model by identifying classification patterns in the training dataset, and the second stage involves the classification of the testing dataset in different class labels [5].

In the learning phase, machine learning classification models give equal importance to each class label in a dataset. However, in reality, the number of samples of each of the class labels may not be the same in any dataset. When the difference between the numbers of these class labels is high, the dataset is referred to as an imbalanced dataset, and this can negatively affect the performance of a prediction model [6]. A prediction model may show high accuracy by correctly classifying the majority class but still misclassify the minority class. The cost of misclassifying the minority class can lead to serious consequences. In real life, balanced datasets are rare, and models trained on imbalanced datasets may have to pay high cost of a misclassification. To address the issue of neglecting the minority class in imbalanced datasets, the input dataset can either be upsampled using the Synthetic Minority Oversampling Technique (SMOTE), or appropriate evaluation metrics such as Matthews Correlation Coefficient (MCC) and macro F1-score can be used to assess model performance more accurately [4,7].

# Materials And Methods

## Literature review

Shivprasad et al. [8] conducted a research "The ORNATE India Project," funded by UK Research and Innovation (UKRI) through the Global Challenges Research Fund. The objective of this research was to examine the impact of diabetes on diabetes-related visual impairment. Zaman et al. [9] proposed a diabetes classification model using Support Vector Machine (SVM), Naïve Bayes, Random Forest (RF), and Decision Tree (DT). They used the PIMA Diabetes dataset and achieved an accuracy of 81% with the Naïve Bayes classifier.

Handling imbalanced datasets is a major challenge in the accurate classification of medical data. Incorrect classifications can lead to serious consequences for patients, potentially costing them their lives. To increase minority class samples and reduce noise in the dataset, Xu et al. [10] proposed a data sampling model called RFMSE, which is the combination of the misclassification-oriented Synthetic Minority Over-sampling Techniqu with the Edited Nearest Neighbor method based on RF. The method was applied to 10 UCI datasets, including the diabetes dataset, and algorithm performance was compared using F-value and MCC.

Traymbak and Issar[11] developed a prediction model using the PIMA diabetes dataset implemented in R language. They used five machine learning algorithms - Linear Discriminant Analysis, k-Nearest Neighbor (KNN), SVM, RF, and AdaBoost - and evaluated the model's performance using receiver operating characteristic curve, accuracy, precision, and recall. They found SVM classifier as the best classifier for their model. Viloria et al. [12] conducted a study on a three-class diabetes dataset collected from approximately 500 Colombian patients. They implemented an SVM classifier, achieving an accuracy of 99.2% on that dataset.

Tigga and Garg [13] implemented a model that used six machine learning algorithms - KNN, LR, SVM, DT, naïve Bayes, and RF - and applied it to two distinct datasets. The first dataset was a primary dataset, containing 952 samples collected via online and offline modes, while the second was the PIMA Diabetes dataset. The model's performance was evaluated using prominent metrics such as accuracy, F1-score, MCC, and cross-validation. RF emerged as the best-performing classifier across both datasets.

## Working methodology

Today in healthcare domain, a number of models based on machine learning and AI techniques are developed to classify or predict outcomes from medical data related to various diseases. To make it more effective, research continues to refine these models. In this paper, a prediction model is proposed on the PIMA diabetes dataset [14] to classify samples of pregnant women into diabetic and non-diabetic categories. Although the size of the dataset is small, efforts have been made to increase the efficiency of the model by using proper preprocessing method, applying feature selection, and tuning the hyperparameter of classifiers. Sometimes a model may achieve high accuracy by correctly predicting only the majority class, while for the minority class, the same may not happen. Therefore, accuracy cannot be considered the sole evaluation metric [15,16] due to the importance of minority class in an imbalanced dataset. The ambiguity exists in the performance of the model, and this can be reduced by either balancing the class labels through oversampling or downsampling techniques or by using appropriate evaluation metrics such as MCC and macro or micro F1-score [7,17,18].

The model is implemented using the Python programming language and its supporting libraries. The framework of the proposed model is illustrated in Figure *1*.

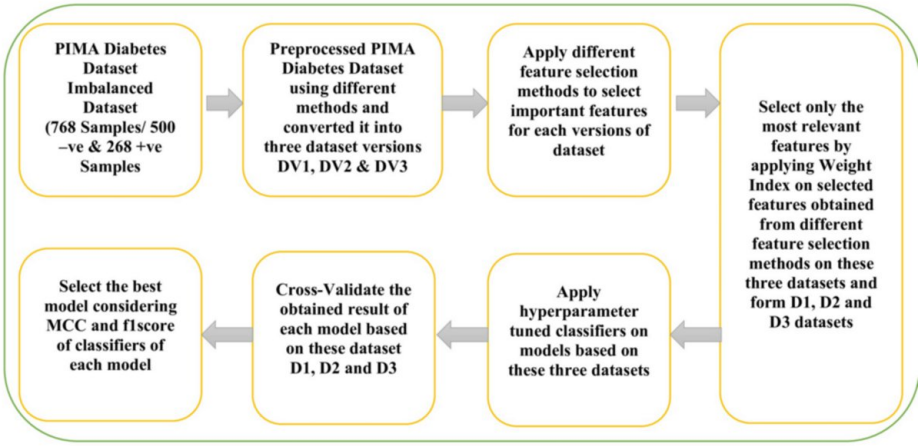


**FIGURE 1: Working methodology of diabetes prediction model**

MCC, Matthews Correlation Coefficient

In the proposed analysis model, three versions of the PIMA dataset were created by applying different preprocessing methods. Furthermore, ExtraTreeClassifier(), SelectKBest(), and some other methods were applied. Since different methods may select varying sets of features, a method called get_weighted_feature_index() is proposed to identify the most relevant features whose weight index is above the given threshold. Finally, three versions of dataset - D1, D2, and D3 - each containing only the highly weighted features - were created. Furthermore, prediction models were implemented using hyperparameter-tuned classifiers on datasets D1, D2 and D3 to analyze the performance. Each step of working methodology of the model has been discussed in detail in the subsequent sections.

*Load Diabetes Dataset*

The research work utilizes the PIMA Indian Diabetes Dataset, which is publicly available on Kaggle [14]. This dataset is a binary class dataset having eight features: glucose, pregnancy, insulin, blood pressure, skin thickness, age, BMI, and DiabetesPedigreeFunction. It contains 500 non-diabetic samples (negative samples indicated by "0") and 268 diabetic samples (positive samples indicated by "1"), so it is a classic example of an imbalanced dataset. Table *1* provides a detailed description of the PIMA dataset. In the dataset, the value of attributes (Pregnancy, Glucose, BloodPressure, skinthickness, Insulin and BMI) for some samples are "0", which may have resulted from missing data during collection or typographical errors. The performance of a model may be adversely affected by these zero values present in the dataset, so preprocessing is required to clean the data before building a model on it.

*Dataset Preprocessing*

The construction of a building cannot be performed on a weak foundation. Similarly, the performance of a prediction model cannot be accurate if it is based on a dataset that is imbalanced and contains noisy data. Six features of the PIMA dataset, pregnancy, glucose, blood pressure, skin thickness, insulin, and BMI, contain zero values for some samples. The number of such samples for these features are 111, 5, 35, 227, 374, and 11, respectively. In medical science, blood pressure, insulin level, or glucose level cannot be zero under any circumstances for a person. This indicates that the PIMA dataset contains noisy data, which must be eliminated before building the model. In this research, the actual PIMA dataset has been converted into three dataset versions called DV1, DV2, and DV3. Table *1* provides detailed information about the size and preprocessing operations of these dataset versions.

| Dataset Versions | Preprocessing Activities | Shape of Datasets | Number of Minority and Majority Classes |
|---|---|---|---|
| DV1 Dataset - PIMA Diabetes Dataset | Ignore missing values; Preprocessing is not performed | 768 × 9 | 268/500 |
| DV2 Dataset - Imputed samples | Fill all missing/unknown values by corresponding column's mean value | 768 × 9 | 268/500 |
| DV3 Dataset - Dataset removing outliers and rows having missing values | Remove rows having missing values and outliers | 337 × 9 | 98/239 |

**TABLE 1: Size and number of majority/minority class of PIMA dataset versions after preprocessing operations**

It is usually uncommon to find a dataset with an equal number of samples for all class labels. The given PIMA dataset is a binary-class dataset with 500 samples in the majority class and 268 samples in the minority class. The dataset versions created through preprocessing operations on the PIMA dataset are also imbalanced [19]. The numbers of majority and minority class samples in these dataset versions are provided in Table *1*.

*Apply Feature Selection*

A dataset contains a number of attributes and two or more class labels. The values of the class labels are determined by all or a subset of the available features. Some features play an important role in the prediction model to decide the class, while others are less important. Using all features in machine learning may degrade classifier performance, require additional computational resources, and increase prediction costs [20]. Feature selection is the process of identifying relevant features and removing irrelevant ones from the dataset. Feature selection methods are classified into four types: filter, wrapper, embedded, and hybrid methods [21]. Wrapper methods build a number of models using different subsets of features and select only those features for which the model produces the best results. Filter methods use a ranking approach to decide the relevance of dataset attributes. They do not tune the model and thus require low computation time. Embedded feature selection methods combine the advantages of both wrapper and filter methods [22].

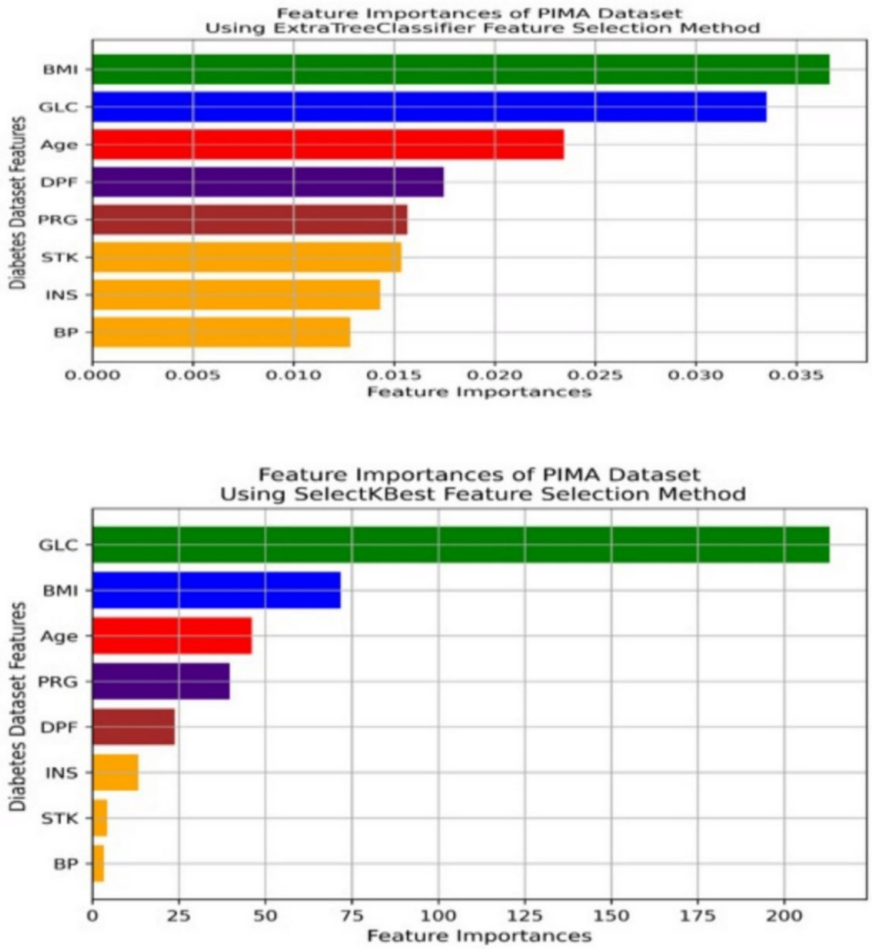


**FIGURE 2: Feature importance of the features of DV1 dataset using ExtraTreeClassifier() and SelectKBest() feature selection method**

DV1 Dataset - Actual PIMA Diabetes Dataset (ignore missing values; preprocessing is not performed)

GLC, Glucose; BMI, Body Mass Index; PRG, Pregnancy; DFP, DiabetesPedigreeFunction; INS, Insulin; STK, Skin Thickness; BP, Blood Pressure

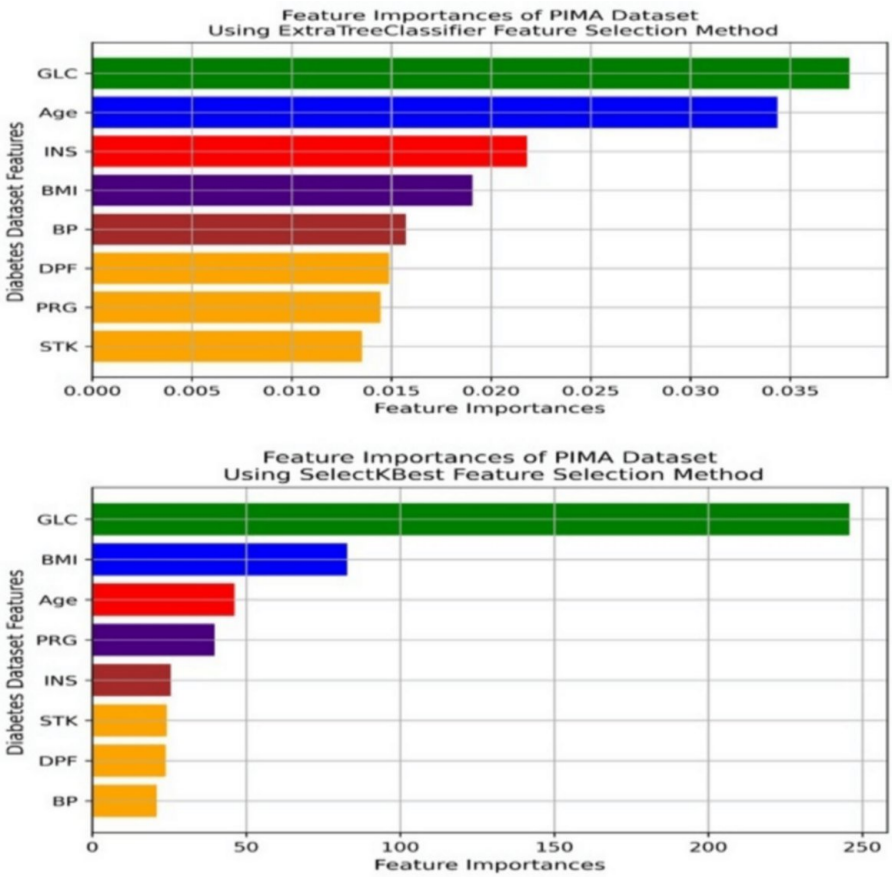


**FIGURE 3: Feature importance of the features of DV2 dataset using ExtraTreeClassifier() and SelectKBest() feature selection method**

DV2 Dataset - Imputed samples (fill all missing/unknown values by corresponding column's mean value)

GLC, Glucose; BMI, Body Mass Index; PRG, Pregnancy; DFP, DiabetesPedigreeFunction; INS, insulin; STK, Skin Thickness; BP, Blood Pressure

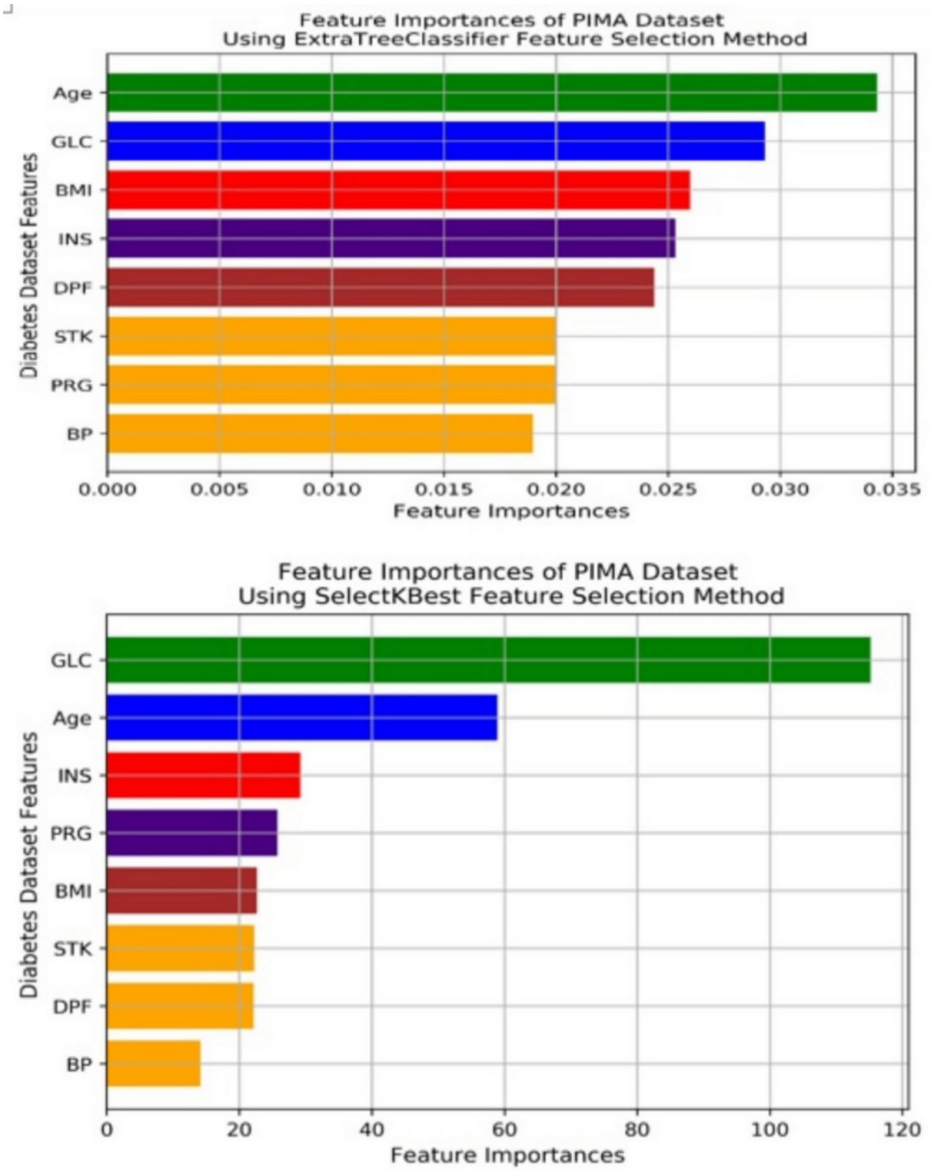


**FIGURE 4: Feature importance of the features of DV2 dataset using ExtraTreeClassifier() and SelectKBest() feature selection method**

DV3 Dataset - Dataset removing outliers and rows having missing values (remove rows having missing values and outliers)

GLC, Glucose; BMI, Body Mass Index; PRG, Pregnancy; DFP, DiabetesPedigreeFunction; INS, insulin; STK, Skin Thickness; BP, Blood Pressure

In the proposed model, a number of feature selection methods, ExtraTreeClassifier (), SelectKBest (), sequentialFeatureSelector (backward), sequentialFeatureSelector (farward), and recursiveFeatureElemination () method, have been applied to identify relevant features from each version on three dataset versions of the PIMA dataset. The feature importance charts for the selected features by ExtraTreesClassifier() and SelectKBest() on dataset versions DV1, DV2, and DV3 are shown in Figures *2*-*4*, respectively. Each method produces a list of selected features. It should be noted that some features are common across the lists returned by all methods. The result produced by the prediction model based only on the common features, available in all selected lists, may not be optimal. Sometimes, there may be no common features among all lists, or the number of common features may be very small, which can lead to less effective results. In this paper, a weighted index model is proposed, based on a threshold value, to select common features from all selected lists. The threshold value defines the acceptability parameter about any specific feature. The range of threshold value lies between 0 and 1.

The threshold value "1" for a feature indicates that the feature must be present in all selected feature lists,

if it is 0.5, it means the feature must be present in half of the list of the selected feature list. The algorithm accepts three inputs and outputs a list of selected features based on the specified threshold value.

The algorithm of the proposed method get_weighted_feature_index() is given below, and its pseudocode is shown in Figure 5.

*Weighted_index (feature_name, selected_features , threshold)*

*{*

*1. Count the occurrence of each feature available in the list of selected features of all feature selection method.*

*2. Calculate the weight of each feature.*

*3. Find the list of features whose weight is greater than or equal to threshold.*

*4. Return list of index of selected features.*

*}*

```
get_weighted_feature_index (features_name, sel_features_indx_list, No_FSM, threshold):
{
        # initialize a counter variable for each feature
1.      feature_count = [0] * len (features_name)
        # takes selected feature's index produced by each feature selection method
        # count the occurrences of a features in different feature's index
2.      for SFL_index in sel_features_indx_list:
3.          do   for feature in features_name:
4.                  if indx[feature] in SFL_index:
5.                          feature_count[indx[feature]] += 1
        # Calculate the weight of features and identify the index of feature having weighted
        # index greater than threshold
6.      weighted_featurs_indx = []
7.      for feature in features_name:
        # Calculate the weight of features
8.          feature_weight = feature_count[indx[feature]] / No_FSM
9.          if feature_weigh >= threshold:                  # if it is greater than
10.             weighted_featurs_indx += indx [ feature ]    # add it to final list
11.     return weighted_featurs_indx
}
```

**FIGURE 5: The pseudocode of get_weighted_feature_index() method**

The inputs for the pseudocode are as follows:

features_name: Name of the features of the PIMA diabetes dataset.

sel_features_indx_list: List of index of the selected features by different feature selection methods.

No_FSM: Number of feature selection methods used in the model.

threshold: A threshold value defines the acceptability parameter.

Output: List of index of selected features according to the given threshold value.

The above algorithm has been applied to the list of selected features from each dataset, and three datasets D1, D2, and D3 have been created. A detailed explanation of the applied algorithm using the input dataset is provided in Table 2. The Get_weighted_feature_index() method is an efficient method to identify the most relevant and reasonable featured dataset. It does not focus on determining the features selected by different selection methods; rather, it emphasizes prioritizing those features that are selected by the majority of the feature selection methods. In other words, a feature selected by most of the methods receives a higher weight compared to features that are selected infrequently. If m is the number of features in the dataset and n feature selection methods are used in the model, the method requires only O(mn) time to produce the index of selected features.

| Input Dataset | Features Selection Method & Selected Features | Index of Selected Feature | Occurrences of Features in Selected Features List | Weighted Index of Features in Selected Features List | Weighted Index of Features ≥Threshold Value (0.75) | Selected Features and Their Index Using Weighted Index() Method |
|---|---|---|---|---|---|---|
| DV1 | ExtraTreeClassifier () | {0,1,2,5,6,7} | {6, 6, 5, 0, 3, 6, 4, 6} | [ 1.00, 1.00, 0.83, 0.00, 0.50, 1.00, 0.67, 1.00] | [ 1.00, 1.00, 0.83, X, X, 1.00, X, 1.00] {0, 1, 2, 5, 7} | [P, G, $B_P$, B, A] {0, 1, 2, 5, 7} |
| | SelectKBest() | {0,1,4,5,6,7} | | | | |
| | Sequential Feature Selector(backward) | {0,1,2,4,5,7} | | | | |
| | Sequential Feature Selector (forward) | {0,1,2,4,5,7} | | | | |
| | Recursive Feature Elimination(DT) | {0,1,2,5,6,7} | | | | |
| | Recursive Feature Elimination (rfc) | {0,1,2,5,6,7} | | | | |
| DV2 | ExtraTreeClassifier () | {0,1,2,5,6,7} | {6, 6, 5, 1, 1, 6, 5, 6} | [ 1.00, 1.00, 0.83, 0.16, 0.16, 1.00, 0.83, 1.00] | [ 1.00, 1.00, 0.83, X, X, 1.00, 0.83, 1.00] {0, 1, 2, 5, 6, 7} | [P, G, $B_P$, B, D, A] {0, 1, 2, 5, 6, 7} |
| | SelectKBest() | {0,1,3,4,5,7} | | | | |
| | Sequential Feature Selector(backward) | {0,1,2,5,6,7} | | | | |
| | Sequential Feature Selector (forward) | {0,1,2,5,6,7} | | | | |
| | Recursive Feature Elimination(DT) | {0,1,2,5,6,7} | | | | |
| | Recursive Feature Elimination (rfc) | {0,1,2,5,6,7} | | | | |
| DV3 | ExtraTreeClassifier () | {0,1,3,4,5,7} | {5, 6, 3, 4, 5, 4, 3, 6} | [0.83, 1.0,0.5, 0.67,0.83,0.67,0.50,1.00] | [0.83, 1.0,X,X,0.83,X,X,1.00] {0, 1, 4, 7} | [P, G, I, A] {0, 1, 4, 7} |
| | SelectKBest() | {0,1,3,4,5,7} | | | | |
| | Sequential Feature Selector(backward) | {0,1,2,3,4,7} | | | | |
| | Sequential Feature Selector (forward) | {0,1,2,4,6,7} | | | | |
| | Recursive Feature Elimination(DT) | {0,1,2,5,6,7} | | | | |
| | Recursive Feature Elimination (rfc) | {1,3,4,5,6,7} | | | | |

**TABLE 2: Demonstration of weighted index () algorithm on selected features list for PIMA datasets**

Features of the PIMA Diabetes Dataset: ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']

Abbreviations of features of the PIMA Diabetes Dataset: [P, G, $B_P$, S, I, B, D, A]

Index of features of the PIMA Diabetes Dataset: { 0, 1, 2, 3, 4, 5, 6, 7 }

*Applied Hyperparameter-Tuned Classifiers*

The classifiers used in machine learning are algorithms that enable a machine to classify the samples of an input dataset. These classifiers, which are implementations of sophisticated statistical and mathematical methods, are used by classification/prediction models to train and test the model. The classifiers chosen for this research work are KNN, SVM, DT, and RF. Each classifier has a number of attributes that determine its behavior - these are called hyperparameters. Hyperparameters are used to optimize the performance of the model. Hyperparameter tuning is the process of finding the set of optimal

hyperparameter values that produce the best results for the model. The hyperparameters that have been tuned, along with their ranges, are given in Table 3.

| S. No. | Classifiers | Hyperparameters of Classifiers | Tested Value of Hyperparameters |
|---|---|---|---|
| 1 | K-nearest neighbor | n_neighbors | Range ( 3, 30) |
| | | metrics | ['euclid' , 'minkowski' ] |
| | | algorithm | ['brute', 'kd_tree', 'ball_tree', 'auto'] |
| | | p | {1, 2} |
| | | leaf_size | {5 ,10,15,20,25,30} |
| 2 | Decision tree | criterion | [ 'entropy' , 'gini' ] |
| | | maximum_depth | Range ( 1 to 50) |
| | | Minimum_sample_size | Range ( 1 to 10) |
| | | random_state | Range 1 to 66 |
| 3 | Random forest | n_estimators | {25, 50, 75, 100 } |
| | | criterion | ['entropy' , 'gini'] |
| | | random_state | Range 1 to 50 |
| | | max_features | 'auto' |
| | | Max_depth | {5,10,15} |
| 4 | Support vector machine | kernel | ['rbf', 'poly', 'linear'] |
| | | C | {25,50,75, 100, 125, 150 , 175, 200, 225, 250} |
| | | gamma | scale' |
| | | degree | {1, 2, 3} applicable for 'poly' kernel |

**TABLE 3: Classifiers and their hyperparameters of the prediction model**

In a prediction model, the input dataset is divided into a training dataset and a test dataset. The training dataset is used for learning purpose and building the model, while the test dataset is used to test the model. In this study, k-fold cross-validation is used to verify the results of the prediction model. Instead of using a fixed portion for training and testing, the k-fold cross-validation method divides the entire dataset into k sections; k-1 sections are used to train the model, and one section is used to test it. This process is repeated k times by changing the sections of training and testing datasets. In this way, the entire dataset is used for both training and testing. The final result of a classifier is obtained by averaging the performance across all k folds [23].

In earlier sections of the paper, the methodology of the research work is discussed in detail and also illustrated in Figure 6. First, the input dataset was processed and cleaned using different preprocessing methods, resulting in three dataset versions called DV1, DV2, and DV3. In the next step, the dimensionality of each dataset was reduced using feature selection methods. A weighted index method was then proposed to identify and select the most relevant features from the list of selected features. The outcome of these activities comes in the form of Dataset D1, Dataset D2, and Dataset D3. Furthermore, hyperparameter-tuned classifiers, along with cross-validation, were applied to build three prediction models based on D1, D2, and D3. The final results were determined by analyzing the results obtained from these model on MCC and F1-score.

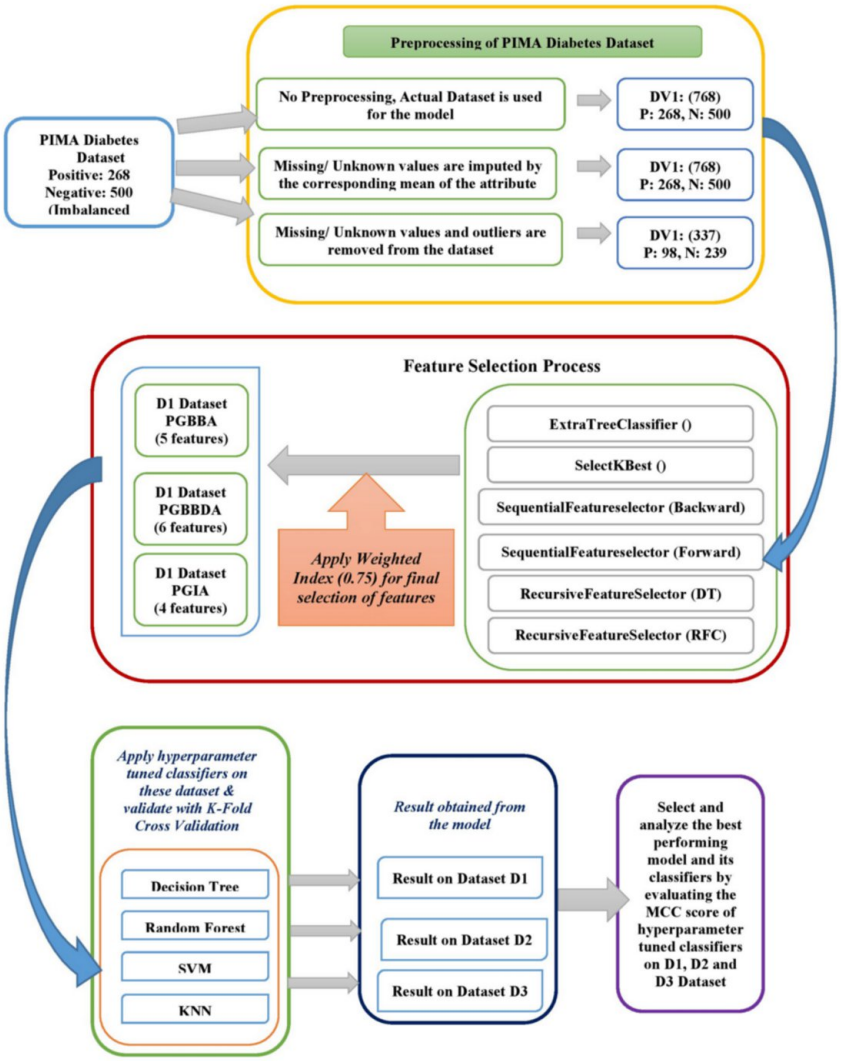


**FIGURE 6: Details of working methodology of the prediction model**

DT, Decision Tree; KNN, K-Nearest Neighbors; RFC, Random Forest Classifier; SVM, Support Vector Machine; MCC, Matthews Correlation Coefficient

# Results

## Experimental setup of the model

In the experimental setup of the model, the PIMA diabetes dataset, which is publicly available on dataset repositories UCI Machine Learning and Kaggle, has been used. The model is implemented in Python using its supporting packages. Matplotlib has been used for visualization of charts and graphs, and sklearn for preprocessing the dataset, applying classifiers, and their hyperparameter tuning.

First, the diabetes dataset has been preprocessed properly in different ways, and from it, three versions of the dataset were created; the details of the preprocessing and the dataset sizes are mentioned in Table *1*. Further, six feature selection methods have been applied to these three versions of the dataset, and each feature selection method selects a different subset of features as the most relevant for each version of the dataset. Some features are common across the different outputs of the selection process. Therefore, a weighted index method is applied to calculate the weight of a selected feature across the outputs of different methods, and the features which pass the threshold value, defined in the model, are finally selected as the features of the dataset for the analysis model. The same process has been applied for each version of the dataset.

In the next step, classification models using six machine learning classifiers have been built on these three datasets, which classify their samples as positive or negative class labels. To optimize the performance of the classifiers, their hyperparameters have also been tuned, and the models have been evaluated on different performance metrics, among which MCC is the most important. In the last step of the research work, the obtained results have been analyzed in detail.

In this section, an analysis is made on the obtained results of models built on datasets D1, D2, and D3. The details of preprocessing methods, feature selection methods, and dataset sizes are given in previous figures and tables. A number of performance metric values, such as MCC, F1-score, accuracy, precision, recall, specificity, and root mean square error (RMSE), have been produced by hyper-tuned classifiers and further validated by K-fold cross-validation [7,16,23]. Since the PIMA dataset is imbalanced, the main focus is given to the metric MCC to decide the best model and classifiers. The formulas used to calculate these metrics are given below, where TP (true positives), TN (true negatives), FP (false positives), and FN (false negatives) values are obtained from the confusion matrix of the classifiers.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Precision = TP/(TP + FP)$$

$$Recall = TP/(TP + FN)$$

$$Specificity = TN/(TP + FN)$$

$$F1score = 2 * (Precision * Recall)/(Precision + Recall)$$

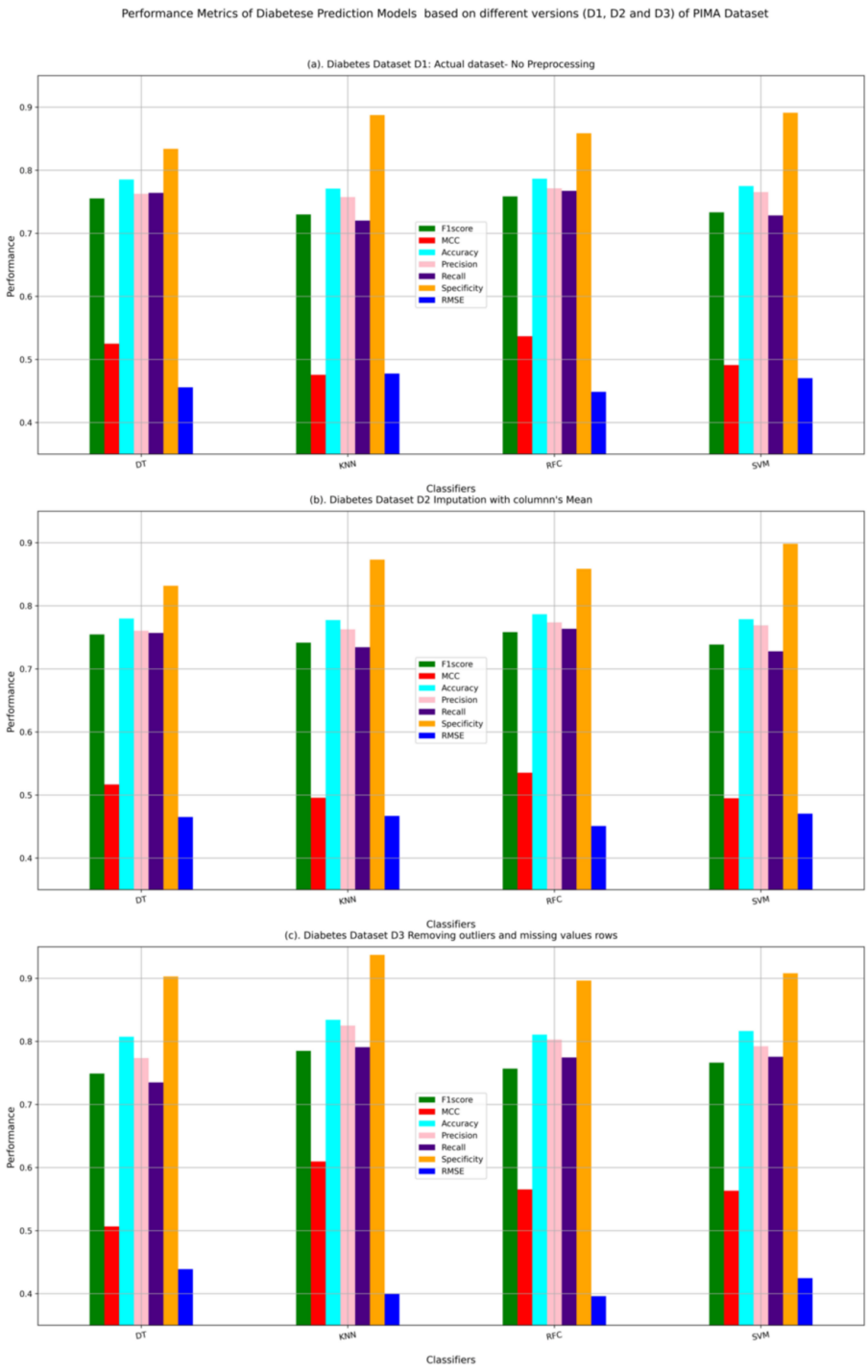


**FIGURE 7: Performance metrics of classifiers of model based on datasets D1, D2, and D3**

DT, Decision Tree; KNN, K-Nearest Neighbors; RFC, Random Forest Classifier; SVM, Support Vector Machine; MCC, Matthews Correlation Coefficient; RMSE, Root Mean Square Error

In Figure 7, a bar chart shows the results of models built on three versions of the PIMA dataset, D1, D2, and D3, in sub-figures 7(a), 7(b), and 7(c), respectively. For the model based on dataset D1, random forest is the best classifier since it shows the maximum scores for MCC, F1-score (macro), accuracy, precision, recall, and the minimum value for error rate. The values of these metrics are 53.68%, 75.85%, 78.65%, 77.12%, 76.72%, and 44.87%, respectively. Random forest shows the above performance at 10-fold cross-validation. The higher values of MCC or F1-score show that the model is performing well, but there is still scope for improvement. A similar result is shown by the model on the D2 dataset, where random forest produces the same result at 10-fold cross-validation with a slight improvement. The best model is built on

dataset D3, as shown in Figure 7(c). Here, the maximum MCC, F1-score (macro), accuracy, precision, recall, and specificity values are 60.95%, 78.5%, 83.42%, 82.49%, 79.08%, and 94.71%, respectively, achieved by the KNN classifier at 7-fold cross-validation when k (number of neighbors) is 9.

# Discussion

## Performance metrics for imbalanced dataset

The performance of a prediction model based on an imbalanced dataset cannot be judged using general metrics such as accuracy, precision, etc. The accuracy predicted by a binary class dataset prediction model is always high if its majority class belongs to the true class. Even if the model has high accuracy, it may not have learned anything from the dataset [16]. These metrics assume that the distribution of majority and minority class labels will be the same in the training and testing datasets; if not, as in real cases, the performance of the prediction model may be misleading. The problems related to imbalanced datasets may be decreased by either making class labels equal (oversampling or downsampling) or using appropriate metrics such as MCC and macro or micro F1-score [8,17,18].

Accuracy is a better metric when the model has to focus on the correct prediction of the positive class; the same may be assumed for the negative class, which is not correct. In such situations, F1-score or MCC are better metrics. MCC is a better metric in comparison to accuracy and F1-score since it considers all four terms of the confusion matrix (TP, TN, FP, and FN). MCC is calculated from the values of TP, TN, FP, and FN taken from the confusion matrix of a classifier.

$$MCC = (TP*TN - FP*FN)/\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}$$

MCC ranges between -1 and +1; the closer the score of a classifier is to +1, the better the performance shown by the classifier. In other words, a value closer to +1 indicates that the classifier predicted both classes very well, even if the input dataset is imbalanced [17,20]. RMSE is applied to evaluate the error made by the models. The error of a model is the difference between the predicted true value by a model and the actual positive values in the dataset. RMSE is a way to measure the error of a prediction model in predicting quantitative data. RMSE squares the errors before taking their average, and thus it counts every error made by the model [18]. This indicates that RMSE is a better metric to measure error in comparison to mean squared error. The formula to calculate RMSE is given below, where $n$ is the total number of values, $d$ is the predicted value, and $p_i$ is the actual observed value.

$$RMSE = \sqrt{(2 * \sum (d - p_i)^2}$$

The best-performing classifiers and their score values on these metrics for these dataset models are summarized in Table 4, which shows that RF is the best-performing classifier for dataset models D1 and D2, while KNN is for model D3.

| Performance Metrics | Dataset $D_1$: Actual PIMA Diabetes Dataset | | Dataset $D_2$: Missing Values Are Filled by Mean | | Dataset $D_3$: Removing Rows Having Missing Values & Outliers | |
|---|---|---|---|---|---|---|
| | Max. Score | BPC | Max. Score | BPC | Max. | BPC |
| MCC | 53.68 | RF | 53.53 | RF | 60.95 | KNN |
| F1-Score | 75.85 | RF | 75.83 | RF | 78.50 | KNN |
| Accuracy | 78.65 | RF | 78.65 | RF | 83.42 | KNN |
| Precision | 77.12 | RF | 77.36 | RF | 82.49 | KNN |
| Recall | 76.73 | RF | 76.36 | RF | 79.08 | KNN |
| Specificity | 90.74 | SVM | 89.83 | SVM | 93.71 | KNN |
| RMSE (Min) | 44.87 | RF | 45.08 | RF | 39.60 | RF |

**TABLE 4: Performance metrics of the model evaluated on datasets D1, D2, and D3, highlighting their best-performing classifiers**

BPC, Best Performing Classifier; MCC, Matthews Correlation Coefficient; RMSE, Root Mean Square Error; RF, Random Forest; KNN, K-Nearest Neighbors; SVM, Support Vector Machine

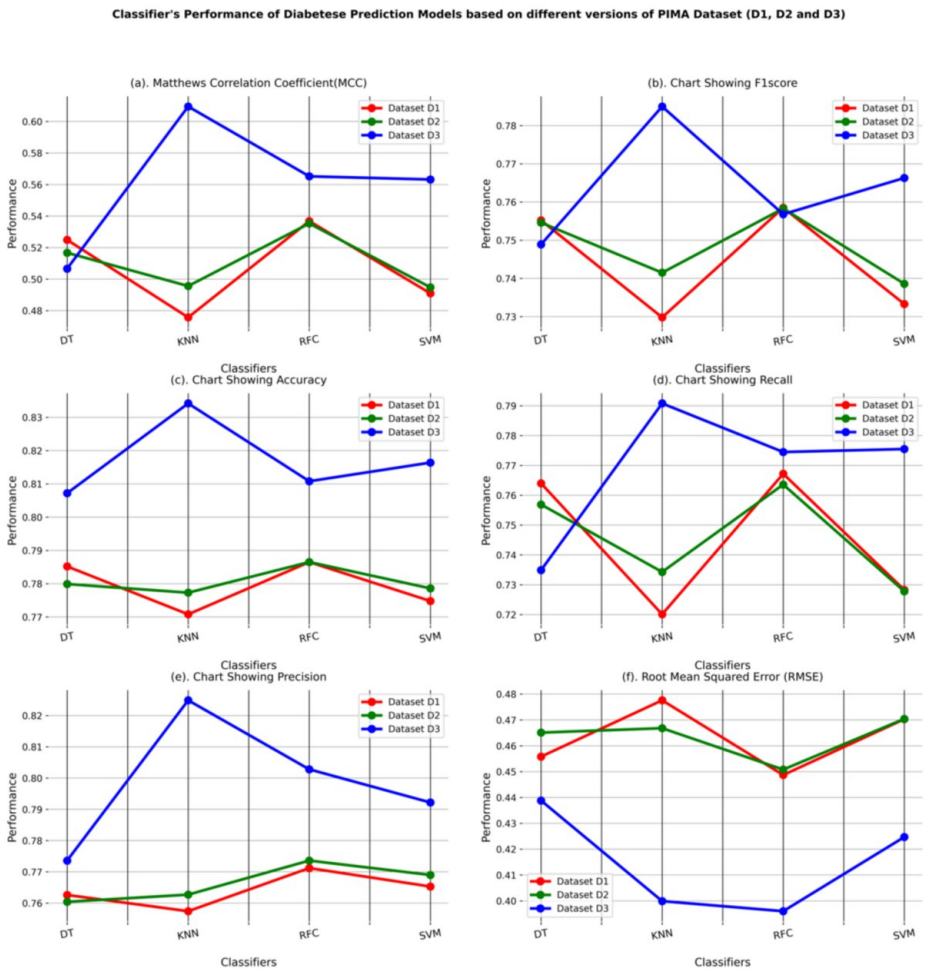


**FIGURE 8: Performance of classifiers across various evaluation metrics**

DT, Decision Tree; KNN, K-Nearest Neighbors; RFC, Random Forest Classifier; SVM, Support Vector Machine; MCC, Matthews Correlation Coefficient; RMSE, Root Mean Square Error

In Figure *8*, the obtained results are shown in the form of a line chart for each dataset model, with each metric represented separately. The chart shows that the performance of the models built on datasets D1 and D2 is somewhat similar, while the performance of the third model, built on the D3 dataset, is significantly better in comparison to the first two across all the metrics used. Although the D3 model produces better results compared to the others, it has only 337 samples, which may not generalize well. This approach may not be effective for datasets containing a large number of samples with missing or unknown values. A smaller sample size may introduce sampling bias. Such problems can be addressed using imputation with uncertainty (e.g., multiple imputation) or synthetic data augmentation (e.g., SMOTE). Since the focus of the article is on the application of MCC as a better metric for imbalanced datasets, synthetic data augmentation is beyond the scope of this research.

## Conclusions

The study used four classifiers to analyze the behavior of prediction models based on different versions of the PIMA diabetes datasets. The first model was used on the actual PIMA dataset without any preprocessing. The second model was built on the imputed version of the input dataset, where all missing values were filled by the mean of the corresponding columns. The third model was built on a cleaned version of the dataset created after the removal of outliers and samples having missing values. Considering the performance of classifiers used in the models, the third model's outcome shows the best performance, followed by the second model, and lastly the first model. From the classifier's point of view, RF is the best-performing classifier for the first and second dataset models, while KNN is the best-performing for the third model. The performances of classifiers have been evaluated considering MCC as the main metric, and all results are validated by K-fold cross-validation.

The study shows that the best model has been built on a dataset without any outliers and after removing all rows with missing values. However, this reduces the size of the PIMA dataset to just 337 samples from

768 samples. This approach may not be fruitful for datasets that contain most samples with missing or unknown values. Removing such samples may not be a good approach for small-sized datasets. The study was conducted on a small-sized dataset. To verify the observations obtained from this research, a model will be built on a larger dataset, and a deep learning approach will be applied to improve its performance.

# Additional Information

## Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

**Concept and design:** Subhash C. Gupta, Noopur Goel

**Acquisition, analysis, or interpretation of data:** Subhash C. Gupta, Noopur Goel

**Drafting of the manuscript:** Subhash C. Gupta, Noopur Goel

**Critical review of the manuscript for important intellectual content:** Subhash C. Gupta, Noopur Goel

**Supervision:** Noopur Goel

## Disclosures

**Human subjects:** All authors have confirmed that this study did not involve human participants or tissue. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

# References


1. International Diabetes Federation: IDF Diabetes Atlas 11th Edition. International Diabetes Federation, Brussels, Belgium; 2025. https://diabetesatlas.org/resources/idf-diabetes-atlas-2025/.
2. World Health Organization: Global Report on Diabetes. WHO Library, Geneva; 2016. https://iris.who.int/bitstream/handle/10665/204871/9789241565257_eng.pdf.
3. Saxena R, Sharma SK, Gupta M, Sampada GC: A comprehensive review of various diabetic prediction models: a literature survey [RETRACTED]. Journal of Healthcare Engineering. 2022, 2022:9814370. 10.1155/2022/8100697
4. Gupta SC, Goel N: Predictive modeling and analytics for diabetes using hyperparameter tuned machine learning techniques. Procedia Computer Science. 2023, 218:1257-1269. 10.1016/j.procs.2023.01.104
5. Kaur H, Kumari V: Predictive modelling and analytics for diabetes using a machine learning approach. Applied Computing and Informatics. 2022, 18:90-100. 10.1016/j.aci.2018.12.004
6. Zhang P, Fonnesbeck C, Schmidt DC, White J, Kleinberg S, Mulvaney SA: Using momentary assessment and machine learning to identify barriers to self-management in type 1 diabetes: observational study. JMIR Mhealth Uhealth. 2022, 10:e21959. 10.2196/21959
7. Halim AM, Dwifebri M, Nhita F: Handling imbalanced data sets using SMOTE and ADASYN to improve classification performance of Ecoli data sets. Building of Informatics, Technology and Science (BITS). 2023, 5:246-253. 10.47065/bits.v5i1.3647
8. Sivaprasad S, Raman R, Conroy D, et al.: The ORNATE India Project: United Kingdom-India Research Collaboration to tackle visual impairment due to diabetic retinopathy. Eye. 2020, 34:1279-1286. 10.1038/s41433-020-0854-8
9. Zaman Z, Shohas M, Bijoy M, Hossain M, Sakib S: Assessing machine learning methods for predicting diabetes among pregnant women. International Journal of Advancement in Life Sciences Research. 2022, 5:29-34. 10.31632/ijalsr.2022.v05i01.005
10. Xu Z, Shen D, Nie T, Kou Y: A hybrid sampling algorithm combining M-SMOTE and ENN based on random forest for medical imbalanced data. Journal of Biomedical Informatics. 2020, 107:103465. 10.1016/j.jbi.2020.103465
11. Traymbak S, Issar N: Data mining algorithms in knowledge management for predicting diabetes after pregnancy by using R. Indian Journal of Computer Science and Engineering. 2021, 12:1542-1558. 10.21817/indjcse/2021/v12i6/211206006
12. Viloria A, Herazo-Beltran Y, Cabrera D, Pineda OB: Diabetes diagnostic prediction using vector support machines. Procedia Computer Science. 2020, 170:376-381. 10.1016/j.procs.2020.03.065
13. Tigga NP, Garg S: Prediction of type 2 diabetes using machine learning classification methods. Procedia Computer Science. 2020, 167:706-716. 10.1016/j.procs.2020.03.336
14. Pima Indians Diabetes Database | Kaggle. Accessed: September 27, 2024: https://www.kaggle.com/uciml/pima-indians-diabetes-database.
15. Gupta SC, Singh DK, Goel N: Enhancing the performance of diabetes prediction using tuning of hyperparameters of classifiers on imbalanced dataset. Indian Journal of Computer Science and Engineering.

2021, 12:1646-1662. 10.21817/indjcse/2021/v12i6/211206049
16. Best Techniques and Metrics for Imbalanced Dataset. Accessed: September 27, 2024: https://www.kaggle.com/code/marcinrutecki/best-techniques-and-metrics-for-imbalanced-dataset.
17. Ferri C, Hernández-Orallo J, Modroiu R: An experimental comparison of performance measures for classification. Pattern Recognition Letters. 2009, 30:27-38. 10.1016/j.patrec.2008.08.010
18. Chicco D, Tötsch N, Jurman G: The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData Mining. 2021, 14:13. 10.1186/s13040-021-00244-z
19. Roy K, Ahmad M, Waqar K, et al.: An enhanced machine learning framework for type 2 diabetes classification using imbalanced data with missing values. Complexity. 2021, 2021:9953314. 10.1155/2021/9953314
20. Zou Q, Qu K, Luo Y, Yin D, Ju Y, Tang H: Predicting diabetes mellitus with machine learning techniques. Frontiers in Genetics. 2018, 9: 515. 10.3389/fgene.2018.00515
21. Deo R, Panigrahi S: Performance assessment of machine learning based models for diabetes prediction. 2019 IEEE Healthcare Innovations and Point of Care Technologies, (HI-POCT), Bethesda, MD, USA. 2019, 147-150. 10.1109/HI-POCT45284.2019.8962811
22. Deberneh HM, Kim I: Prediction of type 2 diabetes based on machine learning algorithm. International Journal of Environmental Research and Public Health. 2021, 18:3317. 10.3390/ijerph18063317
23. How to Fix k-Fold Cross-Validation for Imbalanced Classification. Accessed: March 5, 2025: https://machinelearningmastery.com/cross-validation-for-imbalanced-classification/.