A robust voting approach for diabetes prediction using traditional machine learning techniques

Atik Mahabub^{1,2}

Received: 4 July 2019 / Accepted: 21 November 2019 / Published online: 25 November 2019
© Springer Nature Switzerland AG 2019

Abstract

The noteworthy advances in biotechnology and biomedical sciences have prompted a huge creation of information, for example, high throughput genetic information and clinical data, produced from extensive Electronic Health Records. To this end, utilization of machine learning and data mining techniques in biosciences is by and by crucial and fundamental in endeavors to change cleverly all accessible data into profitable knowledge. Diabetes mellitus is characterized as a gathering of metabolic issue applying critical weight on human health around the world. Broad research in all parts of diabetes (determination, pathophysiology, treatment, and so forth.) has prompted the age of tremendous measures of information. The point of the present examination is to direct an orderly audit of the uses of machine-learning, data mining strategies and instruments in the field of diabetes. The main theme of this work is to provide a system which can prognosticate the diabetes in patients with better accuracy. Here, eleven well-known machine-learning algorithms like Naïve Bayes, K-NN, SVM, Random Forest, Artificial Neural Network, Logistic Regression, Gradient Boosting, Ada Boosting etc. are used for detection of diabetes at an early stage. The evaluations of all the eleven algorithms are examined on various parameters like accuracy, precision, F-measure and recall. After cross-validation and hyper-tuning, the best three machine-learning algorithms are determined and then used in Ensemble Voting Classifier. The experimental results affirm that the pointed framework can accomplish to outstanding outcome of almost 86% accuracy of the Pima Indians Diabetes Database.

Keywords Diabetes prediction · Voting Classifier · Machine-learning · Data mining · PIDD

1 Introduction

Classification methodologies are extensively utilized in the therapeutic field for ordering information into various classes as per some obliges nearly an individual classifier. One of such application areas is in the diagnosis and classification of diabetes. Diabetes is a disease which influences the capacity of the body in delivering the hormone insulin, which thus makes the digestion of starch irregular and raise the dimensions of glucose in the blood. In Diabetes an individual by and large experiences high glucose [1].

Heighten thirst, intensify hunger and frequent pee are a portion of the side effects caused because of high glucose. Diabetes is inspected as a fundamental genuine wellbeing matter amid which the proportion of sugar substance can't be controlled. Diabetes is not just influenced by different components like tallness, weight, genetic factor and insulin however the real reason considered is sugar focus among all elements. The early recognizable proof is the main solution for remain far from the intricacies [2].

Numerous analysts are directing examinations for diagnosing the maladies utilizing different grouping

✉ Atik Mahabub, atikmahabub1209042@gmail.com | ¹Department of Electronics and Communication Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh. ²Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada.

calculations of machine learning approaches like J48, Support Vector Machine (SVM), Naive Bayes, Decision Tree, Ada Boosting and so forth [3–6]. Information Mining [7–9] and Machine learning (ML) calculations gain its quality because of the ability of dealing with a lot of information to join information from a few distinct sources and coordinating the foundation data in the examination.

This examination work is centered around pregnant ladies experiencing diabetes. In this work, Naive Bayes, SVM, and Decision Tree machine learning grouping calculations are utilized and assessed on the Pima Indians Diabetes Database (PIDD) dataset to discover the expectation of diabetes in a patient. Test execution of all these three calculations is looked at on different measures like BMI, blood pressure, glucose etc. and accomplished great precision.

The whole work is presented on four sections as follows. Sect. 2 describes the related works in the field of diabetes prediction. A review about likelihood classifications and algorithm are talked about in Sect. 3. Section 4 presents the experimental results. In this part, further discussions and analyzations are also presented. Section 5 is the brief summary of this work and the blueprint of the future works.

2 Literature survey

Perveen et al. [9] talked about the job of Ada Boost and Bagging machine learning techniques utilizing J48 decision tree to characterize the Diabetes Mellitus and patients considering diabetes hazard factors. Results accomplished after the analysis demonstrated that Ada Boost machine learning ensemble system beats well similar bagging just as a J48 choice tree. Kumar et al. [10] used multilayer perceptron (MLP) and Bayes net classifiers, where MLP gave the highest accuracy for the PIDD dataset.

Esposito et al. [11] and Orabi et al. [12] structured a framework for diabetes forecast depending on the idea of machine learning, by applying decision tree. The fundamental point of Orabi et al. is the expectation of diabetes at a specific age and demonstrated the higher exactness in foreseeing the diabetes episodes.

Bashir et al. [13] presented Hierarchical Multi-level classifiers bagging with Multi-objective upgraded Voting (HM-Bag Moov) procedure to classify diabetes and contrasted with different strategies such as Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), Quadratic discriminant analysis (QDA), K-Nearest Neighbors (k-NN), Random Forest (RF) and Artificial Neural Network (ANN). However, the work did not consider hyper-tuning and cross-validation techniques and used limited number of

ML algorithms to ensemble. Finally, HM-Bag Moov Voting Classifier demonstrated an accuracy of 77.21%.

Furthermore, various calculations and various methodologies have been connected, for example, conventional ML calculations, ensemble learning approaches and affiliation standard learning to accomplish the best classification accuracy. Malik et al. [14] has recently compared different calculations of LR, SVM and ANN with the special feature of three-fold cross-validation. It was demonstrated that SVM provides more accuracy than the others.

Meraj Nabi et al. [15] connected four unique classifiers, for example, NB, LR, J48, and RF, and observed the best exactness of 80.43% utilizing LR. As of late, Suri's group (Maniruzzaman et al. [16]) additionally connected four unique classifiers, for example, LDA, QDA, NB, and GPC. The work demonstrated that GPC based spiral premise piece gave the most elevated grouping precision (~ 82%) as for others. It turns out to be helpful for diabetes expectation requiring little to no effort.

Rashid et al. [17] structured an expectation show with two sub-modules, ANN (Artificial Neural Network) was utilized in the principal module and FBS (Fasting Blood Sugar) was utilized in the second module, to foresee diabetes-endless infection. Decision Tree (DT) was used to recognize the side effects of diabetes on patients' wellbeing.

Nai-arun et al. [7] connected a calculation which characterizes the danger of diabetes mellitus. To satisfy the target, four famous machine learning characterization techniques specifically Decision Tree, Artificial Neural Systems, Logistic Regression and Naive Bayes were explored. For improving the heartiness of structured model Bagging and Boosting methods are utilized. Experimentation results demonstrated that the Random Forest calculation gives ideal outcomes among every one of the methods utilized.

Sisodia et al. [18] predicted diabetes on the basis of Naïve Bayes, Decision Tree and SVM at the PIDD Dataset. The work is performed on WEKA software. The results showed that Naïve Bayes works much better than the other two classifiers. The best accuracy is provided during this work is 76.30% by Naive Bayes.

In this work, Ensemble Voting Classifier (EVC) has been used with the PIDD dataset. Ensemble Voting Classifier (EVC) is one of the Machine Learning (ML) algorithms, which is the mixture of different ML algorithms. Here, Ensemble Voting Classifier has been used to get maximum output from the top performed ML classifiers.

3 Methodology

The proposed ensemble architecture can be divided into several sections. The flow diagram of each section can be seen in Fig. 1. The whole work has been implemented in

Fig. 1 Flowchart of the proposed Ensemble Voting algorithm architecture

Python Note-Book version 3.7.6 The various toolkits like pandas, NumPy, matplotlib and scikit-learn have been used to perform different tasks in different subsections. The work done in each section and the corresponding functions used from python toolkits are explained below.

3.1 PIDD dataset

Pima Indians Diabetes Database (PIDD) is very well-known dataset for diabetes prediction works [19]. The dataset has 9 columns and 768 rows. The columns are categorized according to Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age and Outcome. Outcome provides the results of the patients if he/she has diabetes or not. From the panda's library, read function has been used. In particular, read_csv function has been utilized since the dataset has been in excel format in this work.

3.2 Visualization of data information

In this section, the data will be represented, where it would show the pie chart of the percentage of how many patients are affected by diabetes. Apart from that, it would present the information of data like Pregnancies, Glucose, Blood Pressure, Insulin, Age etc. It would also represent how many patients are affected by diabetes among the 768 people. The functions of pyplot, plot, axis etc. of matplotlib toolkit has been basically used for the graphical representation of outputs.

3.3 Preprocessing

The real data index, which comprises of numerous mis-steps, needed to be refreshed and expelled so as to have exact results of the data index. In this progression data collection, it is changed, normalized and coordinated into a proper arrangement before classifiers are connected in the data index like finding out missing data, deleting unnecessary columns etc. The record is properly handled before classifiers are connected on it.

3.4 Machine-learning algorithms

Subsequent to having the preprocessed data, the known machine learning classifiers have been used from the scikit learn toolkit of Python. Scikit-learn is a simple and efficient toolkit for data mining and data analysis. In this work, this toolkit has been mostly used. Firstly, from the model_selection function, 'train_test_split' has been used to split or divide the dataset into training dataset and test dataset. Because of the limited resources of dataset, about 90% dataset has been used for training and 10% dataset has been used for test on random basis. Then the different types of eleven machine learning classifiers are adapted from their corresponding functions to detect diabetes. In particular, Random Forest, Ada Boost, Gradient Boosting, and Voting Classifier have been taken from the ensemble function. As their types, the others like Logistic Regression is taken from the linear_model function, Multi-layer Perceptron (MLP) Classifier is taken from the neural_network function, Decision Tree Classifier is from tree function, Multinomial Naïve Bayes (MultinomialNB)

and Gaussian Naïve Bayes (GaussianNB) are from `naïve_bayes` function, Support Vector Classifier (SVC) is from `svm` function, K-Neighbors is taken from `neighbors` function, and Extreme Gradient Boosting Classifier (XGBClassifier) is taken from `xgboost` function. They are adapted as per their simplicity and popularity. Many works can be found in literature explaining the features and algorithms of these classifier [9, 16, 18]. Since this work focuses on the Voting Classifier, the Ensemble Voting Classifier will only be explained in the following section.

3.5 Hyper-tuning

Hyper parameter improvement or tuning is the issue of picking a lot of ideal hyper parameters for a machine learning algorithm. A hyperparameter is a parameter whose esteem is utilized to control the learning procedure. A similar sort of machine-learning model can require distinctive limitations, loads or learning rates to sum up various information designs [20]. These measures are called hyperparameters and must be tuned with the goal that the model can ideally tackle the machine-learning issue. In this section, Hyper-Tuning would be utilized to get optimum results from the above ML algorithms.

3.6 Cross-validation

This procedure of choosing whether the numerical outcomes evaluating conjectured connections between factors, are worthy as depictions of the information, is known as cross-validation. Here the K-fold cross validation has been performed. Accordingly, the dataset is divided into 10 K-fold.

In implementation of this step, `model_selection` function of `scikit-learn` has been used. Stratified K-Fold sub-function has been used to split the training dataset in K-fold for cross-validation, `cross_val_score` sub-function has been used to observe the cross-validation scores of ML classifiers and `GridSearchCV` sub-function has been used to hyper-tune the ML classifiers.

3.7 Comparisons

This section will compare the eleven ML algorithms with each other based on accuracy after the evaluation and the performance which have been produced from Hyper-tuning and Cross-validations.

3.8 Choosing 3-best classifier

After the performance evaluation of the mentioned well-known classifiers, top three best classifiers have been

identified. Then these top three classifiers are used in the next step to ensemble.

3.9 Utilizing Voting Classifier

For the Voting Approach, the Ensemble Voting Classifier has been chosen here. Top three classifier which are identified previously have been utilized for these Voting Classification to get the best performance and output. Here, top three classifiers have been chosen because more than three classifiers will increase the complexity without significant improvement of results and less than three would compromise with the performance.

3.9.1 Ensemble Voting Classifier

The Ensemble Voting Classifier [21, 22] is a meta classifier for consolidating comparative or adroitly extraordinary machine learning classifiers for classification and detection. The Ensemble Voting Classifier executes “hard” and “soft” voting.

3.9.2 Hard Voting

The hard ensemble voting is the easiest case of majority voting. Here, the class label Y is determined through majority voting of each classifier C_j :

$$Y = \text{mode}\{C_1(x), C_2(x) \dots, C_m(x)\} [j = 1, 2, 3 \dots, m]$$

3.9.3 Soft Voting

In soft ensemble voting, the class names are anticipated depending on the anticipated probabilities p of classifier. This methodology is possibly prescribed if the classifiers are very much aligned.

$$Y = \text{argmax}_i \sum_{j=1}^m W_j P_{ij}$$

where W_j is the load that can be doled out to the j th classifier.

3.10 Performance evaluation

At the last step, the performance of the Voting Classifier will be assessed based on execution measurements Like: test score, ROC score, precision score, recall value etc. The results then will be compared with other relevant works for evaluating the results.

4 Short ideas of used methods

For Ensemble Voting Classifier, there are several ML classifiers have been used for cross-validation. The reasons of using these classifiers has been provided below:

4.1 Decision Tree Classifier

One major bit of leeway of the decision tree model is its straightforward nature. Not at all like other decision-making models, the decision tree makes unequivocal every single imaginable other option and follows every option in contrast to its decision in a solitary view, taking into account simple examination among the different other options. The utilization of discrete hubs to indicate client characterized choices, vulnerabilities, and end of procedure loans further lucidity and straightforwardness to the basic leadership process.

4.2 Naïve Bayes

Super simple, just doing a bunch of counts. If the NB conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than discriminative models like Logistic Regression, so need less preparing information. Additionally, regardless of whether the NB presumption doesn't hold, a NB classifier still regularly works superbly practically speaking. A decent wager if need something quick and simple that performs really well. Its primary detriment is that it can't learn communications between highlights.

4.3 Random Forest

Bunches of approaches to regularize the model and don't need to stress as a lot over your highlights being connected. We likewise have a pleasant probabilistic translation, dissimilar to decision trees or SVMs, and we can without much of a stretch update your model to take in new information (utilizing an online gradient descent strategy), again not at all like choice trees or SVMs. Use it on the off chance that you need a probabilistic system (e.g., to effectively modify characterization limits, to state when we're uncertain, or to get certainty interims) or on the off chance that you hope to get additionally preparing information later on that you need to have the option to rapidly fuse into your model.

4.4 Multi-Layer Perceptron

MLP is a piece of Artificial Neural Network or Neural Network. Neural systems are adaptable and can be utilized for both relapse and arrangement issues. Any information which can be caused numeric to can be utilized in the model, as neural system is a scientific model with guess capacities. Neural systems are great to show with nonlinear information with enormous number of contributions; for instance, pictures. It is dependable in a methodology of errands including numerous highlights. It works by parting the issue of order into a layered system of less difficult components. When prepared, the expectations are quick. Neural systems can be prepared with any number of information sources and layers. Neural systems work best with more information focuses.

4.5 Logistic Regression

It is a broadly utilized method since it is proficient, doesn't require an excessive number of computational assets, it's exceptionally interpretable, it doesn't require input features to be scaled, it doesn't require any tuning, it's anything but difficult to regularize, and it yields well-aligned anticipated probabilities. Like straight relapse, calculated relapse improves when you evacuate ascribes that are irrelevant to the yield variable just as traits that are fundamentally the same as (corresponded) to one another. Thusly, Feature Engineering assumes a significant job with respect to the presentation of Logistic and furthermore Linear Regression. Another preferred position of Logistic Regression is that it is amazingly simple to actualize and proficient to prepare. It is normally started with a Logistic Regression model as a benchmark and have a go at utilizing increasingly complex calculations from that point on. In light of its straightforwardness and the way that it tends to be actualized moderately simple and snappy, Logistic Regression is additionally a decent benchmark that you can use to quantify the exhibition of other progressively complex Algorithms.

4.6 Boosting tree

While boosting isn't algorithmically obliged, most boosting calculations comprise of iteratively learning feeble classifiers as for a dispersion and adding them to a last solid classifier. At the point when they are included, they are ordinarily weighted somehow or another that is normally identified with the feeble learners' precision. After a frail learner is included, the information loads are straightened out, known as "re-weighting". Misclassified input information put on a higher weight and models that are characterized effectively shed pounds. Along these lines,

future feeble students center more on the models that past powerless students misclassified. The fundamental variety between many boosting calculations is their technique for weighting preparing information focuses and speculations. There are many boosting algorithms however Ada Boosting, Gradient Boosting and Extreme Gradient Boosting has been talked about.

4.7 SVC

High exactness, pleasant hypothetical certifications in regards to overfitting, and with a suitable portion they can function admirably regardless of whether the information isn't straightly distinguishable in the base component space. Particularly famous in content characterization issues where high-dimensional spaces are the standard. Memory-serious, difficult to decipher, and sort of irritating to run and tune.

4.8 K-neighbors

K-NN calculation is easy to comprehend and similarly simple to actualize. To order the new information point K-NN calculation peruses entire dataset to discover K closest neighbors. K-NN is a non-parametric calculation which means there are suspicions to be met to actualize K-NN. Parametric models like straight relapse has heaps of suspicions to be met by information before it very well may be executed which isn't the situation with K-NN. K-NN doesn't expressly manufacture any model, it just labels the new information passage based gaining from verifiable information. New information passage would be labeled with lion's share class in the closest neighbor. Given it's a case based learning; k-NN is a memory-based methodology. The classifier quickly adjusts as we gather new preparing information. It enables the calculation to react rapidly to changes in the contribution during continuous use. The majority of the classifier calculations are anything but difficult to execute for double issues and needs exertion to actualize for multi class though K-NN acclimate to multi class with no additional endeavors.

5 Result analysis

From the collected PIDD dataset containing about 768 patients among which about 268 patients have diabetes and 500 patients do not have diabetes. The representation of the pie chart of this Data set has been provided in Fig. 2.

After preprocessing the dataset, the training data has been split into tenfolds [23]. Then, the eleven different ML algorithms have been hyper-tuned and cross validated to get the optimum results from them. As mentioned

Fig. 2 The pie chart representation of dataset

previously, the eleven algorithms are (a) K-Neighbors, (b) Ada Boost, (c) Decision Tree, (d) Random Forest, (e) Support Vector Machine (SVM), (f) Gradient Boosting, (g) Logistic Regression, (h) Multi-Layer Perceptron (MLP), (i) Multinomial Naïve Bayes, (j) Extreme Gradient Boosting, and (k) Gaussian Naïve Bayes. The tuning parameters and the obtained results of each algorithms are explained below.

5.1 K-neighbors

For K-neighbors classifier, the parameters based on algorithm, leaf size, n-neighbors and weights are tuned to achieve optimized results. After tuning the parameters are found as: 'algorithm': 'auto', 'leaf size': 1, 'n-neighbors': 15, and 'weights': 'uniform'. The corresponding results are accuracy 81.82%, precision 81%, recall 82%, F-1 score 82% and ROC score 77.67%.

5.2 Ada Boost

The parameters tuned are based on learning_rate, n_estimators and random_state. After tuning the best results are obtained for these parameters at 'learning_rate': 0.02, 'n_estimators': 1000, 'random_state': 0. For Ada Boost, the optimum results are accuracy 76.62%, precision 77%, recall 77%, F-1 score 77% and ROC score 72.76%.

5.3 Decision Tree

The optimum results obtained for this classifier tuning the parameters of maximum features, minimum samples leaf, minimum samples split and random state. Thus, the values of these parameters are chosen as: 'maximum features': 'log2', 'minimum samples leaf': 12, 'minimum samples split': 5, 'random state': 0. The corresponding performance are obtained as accuracy 77.92%, precision 77%, recall 78%, F-1 score 77% and ROC score 72.56%.

5.4 Random Forest

For Random Forest classifier, the tuning parameters and their optimized values are: 'bootstrap': False, 'criterion': 'gini', 'maximum depth': None, 'maximum features': 'log2', 'minimum samples leaf': 1, 'minimum samples split': 6, 'n estimators': 100. The corresponding performances are accuracy 75.32%, precision 76%, recall 75%, F-1 score 76% and ROC score 72.96%.

5.5 Support Vector Machine

For SVC, the optimum performance is achieved by tuning C, gamma and kernel [20]. After tuning, these parameters are found as: 'C': 1, 'gamma': 0.0001, 'kernel': 'rbf'. For SVC, accuracy 83.12%, precision 83%, recall 83%, F-1 score 82% and ROC score 76.34%.

5.6 Gradient Boosting

The tuning parameters for Gradient Boosting are learning rate, maximum depth, minimum samples leaf, and n estimators. Their corresponding values are: 'learning rate': 0.01, 'maximum depth': 7, 'minimum samples leaf': 12, and 'n estimators': 200, which produce the optimum results as accuracy 76.62%, precision 77%, recall 77%, F-1 score 77% and ROC score 72.52%.

5.7 Logistic Regression

The parameters tuned in Logistic Regression are based on penalty, tolerance, solver, C, intercept scaling, verbose and maximum iterations. After tuning, the parameters are optimized as: 'C': 100, 'intercept scaling': 2, 'maximum iteration': 100, 'penalty': 'l2', 'solver': 'liblinear', tolerance: 0.0001, 'verbose': 1. The corresponding performance are accuracy 81.92%, precision 81%, recall 82%, F-1 score 81% and ROC score 76.53%.

5.8 Multilayer Perceptron (MLP)

For MLP, the parameters of alpha, hidden layer size, random state, solver and maximum iterations are tuned. The optimum results are obtained for alpha: 1e-07, hidden_layer_sizes: 12, maximum iteration: 1000, 'random_state': 6, 'solver': 'lbfgs'. The corresponding results are obtained as accuracy 84.42%, precision 84%, recall 84%, F-1 score 84% and ROC score 78.42%.

5.9 Multinomial Naïve Bayes

The best results are achieved in this classifier by tuning alpha with its value of 'alpha': 0.1 and the corresponding

results are accuracy 68.83%, precision 70%, recall 69%, F-1 score 69% and ROC score 65.96%.

5.10 Extreme Gradient Boosting

For Extreme Gradient Boosting (XGB Classifier), the parameters based on gamma, learning rate, maximum depth, minimum samples leaf and n estimators are tuned. The obtained values are 'gamma': 10, 'learning rate': 0.02, 'maximum depth': 5, 'minimum samples leaf': 10, 'n estimators': 20. The optimized results are noted as accuracy 80.52%, precision 80%, recall 81%, F-1 score 80% and ROC score 74.45%.

5.11 Gaussian Naïve Bayes

For Gaussian Naive Bayes classifier, we tune the parameters based on variable smoothing which value was finally selected as 1e-05. The performance in such optimum condition is found as accuracy 80.52%, precision 80%, recall 81%, F-1 score 80% and ROC score 76.73%.

Now, the three best classifiers have been chosen based on accuracy and ROC scores. The comparison of these parameters among different classifier is shown in Fig. 3 and Fig. 4, respectively. The respective values of such performance parameters of all considered methods are listed in Table 1. As observed in Fig. 3 and Table 1, the three best classifiers are K-neighbors, SVC and MLP producing the accuracy of 81.85%, 83.12% and 84.42%, and the ROC score of 77.67, 76.34 and 78.42% respectively. Their corresponding precisions are also higher than remaining other methods.

After selecting the best three ML classification algorithms (MLP, SVC and K-neighbors), they are used in Voting Classifier to get maximum test score. The ultimate test score of Ensemble Voting Classifier is 85.71. The other parameters of this classifier are: Precision = 86%, Recall = 86%, F-1 = 85% and ROC score = 79.36%. Here, 'Hard Voting' type Voting Classifier has been performed.

Fig. 3 Comparisons of different algorithms based on accuracy

Fig. 4 Comparisons of ROC scores of different algorithms **a** K-neighbors Classifier, **b** Ada Boost, **c** Decision Tree, **d** Random Forest, **e** Support Vector Classifier (SVC), **f** Gradient Boosting, **g** Logistic

Regression, **h** MLP, **i** Multinomial Naïve Bayes, **j** X-Gradient Boosting, and **k** Gaussian Naïve Bayes

Tables 2 and 3 present the report of Voting Classifier and confusion matrix, respectively. The Voting Classifier (Table 2) provides better accuracy than all other individual ML classifiers that has been provided in Table 1. It also provides more precision, recall and F-1 values than the other ones and individual scores for diabetes and non-diabetes persons are also provided. From the confusion matrix, the True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN) values are identified which are used to find accuracy, precision, recall, and F-1 parameters of

Voting Classifier. Figure 5 presents the ROC curve of the Voting Classifier. The true positive rate is determined as the quantity of true positives partitioned by the aggregate of the quantity of true positives and the quantity of false negatives. From the analysis of this figure, it is seen that it provides more positive class than the negative class by the comparisons with other traditional ML classifiers.

The whole work is performed in python 3.7.6 version at Python Note-Book. There several toolkit available to perform but from that, the listed tool-kits have been used:

Table 1 Comparisons of eleven different classifiers based on accuracy, precision, recall, F-1, ROC scores

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F-1 (%)	ROC score (%)
K-neighbors	81.85	81	82	82	77.67
Ada Boost	76.62	77	77	77	72.76
Decision Tree	77.92	77	78	77	72.56
Random Forest	75.32	76	75	76	72.96
SVC	83.12	83	83	82	76.34
Gradient Boosting	76.63	77	77	77	72.52
Logistic Regression	81.82	81	82	81	76.53
MLP	84.42	84	84	84	78.42
Multinomial Naïve Bayes	68.83	70	69	69	65.96
X-Gradient Boosting	80.52	80	81	80	74.45
Gaussian Naïve Bayes	80.52	80	81	80	76.73

Table 2 Report for Ensemble Voting Classifier

Type	Test score (%)	Precision (%)	Recall (%)	F-1 (%)	ROC score (%)
Non-Diabetes	85.71	85	96	90	79.36
Diabetes		88	62	73	
Average		86	86	85	

Table 3 Confusion Matrix of Voting Classifier

	Predicted: No	Predicted: Yes
Actual: No	51	9
Actual: Yes	2	15

Fig. 5 ROC curve of Voting Classifier

5.11.1 pandas

pandas is a Python toolkit giving quick, adaptable, and expressive information structures intended to make working with organized and time arrangement information both simple and instinctive. Pandas deal with many

different kinds of data like: tabular data, ordered and unordered data, Arbitrary matrix data etc. From the pandas library, read function has been used. The dataset has been in excel format so read_csv function has been utilized in this work.

5.11.2 numpy

numpy toolkit is basically used for performing mathematical operations. There are many functions like mean, max, average, min etc. has been used. Here, in this work, mean function has been used. For the cross-validation the training dataset has been divided into ten parts. So, mean function has been performed to get the average value of these ten parts for cross-validation scores.

5.11.3 matplotlib

matplotlib toolkit has been basically used for the graphical representation of outputs. Here, from this toolkit pyplot function is only utilized and from that plot, axis, xlabel, ylabel, legend, show sub-functions has been used to represent the 2D graph in a plot view and provide proper axis name to make it understandable to viewers.

5.11.4 scikit-learn

scikit-learn is the most simple and efficient toolkit for data mining and data analysis. In this work, this toolkit has been mostly used. Firstly, from the model selection function, train_test_split has been used to split or divide the dataset into training dataset and test dataset. Because of the limited resources of dataset, about 90% dataset has been used for training and 10% dataset has been used for test and it is randomly split. After that, for the cross-validation step, model_selection function has been performed. From this function, StratifiedKFold sub-function has been used to split the training dataset in K-fold for cross-validation,

Table 4 Comparison of the performance of our proposed method with several relevant literature

SN	Authors	Year	Data size and class	Classifier type	Accuracy (%)
1	Kumar Dewangan and Agrawal [10]	2015	768 Controls: 500 Diabetic: 268	Multilayer Perceptron (MLP) Bayes Net	81.19
2	Malik et al. [14]	2016	175 Healthy: 87 Diabetic: 88	Logistic Regression (LR) Support Vector Machine (SVM) Artificial Neural Network (ANN) with threefold cross-validation	84.09
3	Meraj Nabi et al. [15]	2017	768 Controls: 500 Diabetic: 268	Naïve Bayes (NB) Logistic Regression (LR) J48, and Random Forests (RF)	80.43
4	Maniruzzaman et al. [16]	2017	768 Controls: 500 Diabetic: 268	Linear discriminant analysis (LDA) Quad- ratic discriminant analysis (QDA), NB Gaussian Process Classifier (GPC)	81.97
5	Dipti et al. [18]	2018	768 Controls: 500 Diabetic: 268	NB , SVM, Decision Tree (DT)	76.3
6	Sneha et al. [3]	2019	1500	RF, SVM, NB , DT, KNN	82.3
7	Bashir et al. [13]	2016	768 Controls: 500 Diabetic: 268	NB, SVM, LR, QDA, RF, ANN K-nearest neighbor (k-NN) HM-Bag Moov	77.21
8	Proposed	2019	768 Controls: 500 Diabetic: 268	Ensemble Voting Classifier	85.71

Bold type indicates the classifier giving the highest accuracy

cross_val_score sub-function has been used to observe the cross-validation scores of ML classifiers and GridSearchCV sub-function has been used to hyper-tune the ML classifiers.

Then from the scikit-learn toolkit, the machine learning classifiers have been used. From the ensemble function, Random Forest Classifier, Ada Boost Classifier, Gradient Boosting Classifier, Voting Classifier have been utilized. Others are, from the linear_model function Logistic Regression, from the neural_network function Multi-layer Perception Classifier (MLPClassifier), from tree function Decision Tree Classifier, from naïve_bayes function Multinomial Naïve Bayes (MultinomialNB) and Gaussian Naïve Bayes (GaussianNB), from svm function Support Vector Classifier (SVC) and from xgboost function Extreme Gradient Boosting Classifier (XGBClassifier) have been utilized.

At last, metrics function has been used from this scikit-learn toolkit. From metrics function, classification report has been used to acknowledge about the precision score, recall score and F1 score of the ML classifiers to observe their performance. roc_curve and roc_auc_score sub-functions have been used to get ROC curve and ROC scores for the ML classifiers.

There are several papers in literature on the diagnosis and classification of diabetic patients. Kumar Dewangan and Agrawal [10] used MLP and Bayes net classifiers, where MLP gave the highest accuracy of 81.19%. The dataset

consisted of 8 attributes and 768 patients having 268 diabetes and 500 controls. Malik et al. [14] used LR, SVM and ANN with threefold cross validation for limited data size and attributes, where SVM was found to give higher accuracy (84.09%). Meraj Nabi et al. [15] applied four different classifiers such as NB, LR, J48, RF at the PIDD Dataset and obtained the best accuracy of 80.43% using LR. Recently, Maniruzzaman et al. [16] applied four unique classifiers (LDA, QDA, NB, and GPC) and demonstrated that GPC based spiral premise piece provide the best accuracy (~82%). In other work, Deepti Sisodia et al. [18] showed that NB performs much better (76.30%) than the other two (Decision Tree and SVM). Later, Sneha et al. [3] modified the NB by generating the correlation between the attributes and then consider the data with proper attribute in the classifier. Thus, the work demonstrated the improved accuracy (82.3%) by NB classifier at the different dataset.

Bashir et al. [13] introduced Voting (HM-Bag Moov) technique to classify diabetes and compared to various classifier such as NB, SVM, LR, QDA, KNN, RF and ANN. An accuracy of 77.21% was demonstrated for PIDD dataset using this Voting Classifier (HM-Bag Moov). Based on the above discussion, Table 4 confirms that our proposed Ensemble Voting Classifier provides better prediction with an accuracy of 85.71% compared to others conventional classifier. The accuracy can be further improved with the cost of complexity, for example, incorporating feature

selection techniques in the classifier [24] or using deep learning method [25].

6 Conclusions

Diabetes is known as one of the critical and chronic diseases which causes an increase in blood sugar. Undiagnosed diabetes can increase the risk of cardiac stroke, diabetic nephropathy, brokenness and failure of various organs, particularly the eyes, kidneys and veins. Therefore, the detection of diabetes at its early stage is one of the important real-world medical problems. Machine learning (ML), a computational method for automatic learning from experience and improves the performance, is widely considering for this purpose to make more accurate predictions. The motive of this study is to find a model which can prognosticate the likelihood of diabetes in patients with maximum accuracy. Here, eleven machine learning classification algorithms namely K-neighbors, Ada Boost, Decision Tree, Random Forest, Support Vector Classifier (SVC), Gradient Boosting, Logistic Regression, MLP, Multinomial Naïve Bayes, X-Gradient Boosting, and Gaussian Naïve Bayes are used in this experiment to detect diabetes at an early stage on Pima Indians Diabetes Database (PIDD). After cross-validation and hyper-tuning, the performances of all the eleven algorithms are examined on various measures like Precision, Accuracy, F-Measure, and Recall. The three best classifiers obtained from the results are K-neighbors, SVC and MLP which provides the accuracy of 81.85%, 83.12% and 84.42%, and the ROC score of 77.67, 76.34 and 78.42% respectively. These three machine-learning algorithms were then applied in Ensemble Voting Classifier. Results obtained show that the examined Voting Classifier outperforms comparatively other algorithms with the accuracy of about 86%.

Acknowledgement I want to thank you Prof. Dr. Faruque Hossain from the Department of Electronics and Communication Engineering at Khulna University of Engineering and Technology who helped me throughout the editing and formatting.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Filho EG, Pinheiro PR, Pinheiro MCD, Nunes LC, Gomes LBG (2019) Heterogeneous methodology to support the early diagnosis of gestational diabetes. *IEEE Access* 7:67190–67199
- Vijayan VV, Anjali C (2015) Prediction and diagnosis of diabetes mellitus—a machine learning approach. In: 2015 IEEE recent advances in intelligent computational systems (RAICS), pp 122–127. <https://doi.org/10.1109/raics.2015.7488400>
- Sneha N, Gangil T (2019) Analysis of diabetes mellitus for early prediction using optimal features selection. *J Big Data* 6:13. <https://doi.org/10.1186/s40537-019-0175-6>
- Kavakiotis I, Tsave O, Salifoglou A, Maglaveras N, Vlahavas I, Chouvarda I (2017) Machine learning and data mining methods in diabetes research. *Comput Struct Biotechnol J* 15:104–116. <https://doi.org/10.1016/j.csbj.2016.12.005>
- Kanchan BD, Kishor MM (2016) Study of machine learning algorithms for special disease prediction using principal of component analysis. In: 2016 international conference on global trends in signal processing, information computing and communication, IEEE explore. <https://doi.org/10.1109/icgtspicc.2016.7955260>
- Batra M, Agrawal R (2018) Comparative analysis of decision tree algorithms, vol 652. In: Panigrahi B, Hoda M, Sharma V, Goel S (eds) *Nature inspired computing. Advances in intelligent systems and computing*. Springer, Singapore, pp 31–36. https://doi.org/10.1007/978-981-10-6747-1_4
- Nai-arun N, Moungrmai R (2015) Comparison of classifiers for the risk of diabetes prediction. *Proc Comput Sci* 69:132–142
- Fatima M, Pasha M (2017) Survey of machine learning algorithms for disease diagnostic. *J Intell Learn Syst Appl* 09:1–16. <https://doi.org/10.4236/jilsa.2017.91001>
- Perveen S, Shahbaz M, Guergachi A, Keshavjee K (2016) Performance analysis of data mining classification techniques to predict diabetes. *Proc Comput Sci* 82:115–121. <https://doi.org/10.1016/j.procs.2016.04.016>
- Kumar Dewangan A, Agrawal P (2015) Classification of diabetes mellitus using machine learning techniques. *Int J Eng Appl Sci* 2(5):145–148
- Esposito F, Malerba D, Semeraro G, Kay J (1997) A comparative analysis of methods for pruning decision trees. *IEEE Trans Pattern Anal Mach Intell* 19:476–491. <https://doi.org/10.1109/34.589207>
- Orabi KM, Kamal YM, Rabah TM (2016) Early predictive system for diabetes mellitus disease. In: *Industrial conference on data mining*, Springer, pp 420–427. https://doi.org/10.1007/978-3-319-41561-1_31
- Bashir S, Qamar U, Khan FH (2016) IntelliHealth: a medical decision support application using a novel weighted multi-layer classifier ensemble framework. *J Biomed Inform* 59:185–200. <https://doi.org/10.1016/j.jbi.2015.12.001>
- Malik S, Khadgawat R, Anand S, Gupta S (2016) Non-invasive detection of fasting blood glucose level via electrochemical measurement of saliva. *Springerplus* 5(1):701. <https://doi.org/10.1186/s40064-016-2339-6>
- Nabi M, Wahid A, Kumar P (2017) Performance analysis of classification algorithms in predicting diabetes. *Int J Adv Res Comput Sci* 8(3):456–461
- Maniruzzaman M, Kumar N, Abedin MM, Islam MS, Suri HS, El-Baz AS, Suri JS (2017) Comparative approaches for classification of diabetes mellitus data: machine learning paradigm. *Comput Methods Programs Biomed* 152:23–34. <https://doi.org/10.1016/j.cmpb.2017.09.004>
- Rashid TA, Abdullah SM, Abdullah RM (2016) An intelligent approach for diabetes classification, prediction and description. *Adv Intell Syst Comput* 424:323–335. <https://doi.org/10.1007/978-3-319-28031-8>
- Sisodia D, Sisodia DS (2018) Prediction of diabetes using classification algorithms. *Proc Comput Sci* 132:1578–1585
- Pima Indians Diabetes Database. <https://www.kaggle.com/uciml/pima-indians-diabetes-database>
- Candelieri A, Giordani I, Archetti F, Barkalov K, Meyerov I, Polovinkin A, Sysoyev A, Zolotykh N (2019) Tuning

- hyperparameters of a SVM-based water demand forecasting system through parallel global optimization. *Comput Oper Res* 106:202–209
21. Mahabub A, Mahmud MI, Hossain MF (2019) A robust system for message filtering using an ensemble machine learning supervised approach. *ICIC Express Lett Part B Appl* 10:805–811. <https://doi.org/10.24507/icicelb.10.09.805>
22. Raschka S (2015) Python machine learning, chapter 7: combining different models for ensemble learning. Packt Publishing Ltd, Birmingham, pp 40–44
23. Malik MZ, Nawaz M, Mustafa N, Siddiqui JH (2018) Search based code generation for machine learning programs. arXiv e-print archive. Cornell University. arXiv: 1801.09373
24. Maniruzzaman M, Rahman MJ, Al-Mehedi Hasan M, Suri HS, Abedin MM, El-Baz A, Suri JS (2018) Accurate diabetes risk stratification using machine learning: role of missing value and outliers. *J Med Syst* 42:92. <https://doi.org/10.1007/s10916-018-0940-7>
25. Swapna G, Vinayakumar R, Soman KP (2018) Diabetes detection using deep learning algorithms. *ICT Express* 4:243–246. <https://doi.org/10.1016/j.icte.2018.10.005>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.