# Prediction of women's diabetes based on symptoms using machine learning algorithms

Moh. Shohanur Rahman
*department of computer science and engineering*
*BRAC University*
Dhaka, Bangladesh
moh.shohanur.rahman@g.bracu.ac.bd

Saib Ahmed
*department of computer science and engineering*
*BRAC University*
Dhaka, Bangladesh
saib.ahmed@g.bracu.ac.bd

Annajiat Alim Rasel
*department of computer science and engineering*
*BRAC University*
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*—Diabetes is a refractory condition that could lead to a global health care disaster. The detrimental effects of diabetes are currently affecting a sizeable section of the population worldwide, and many of these individuals especially women are not being properly diagnosed. Diabetes may eventually develop major health issues such as nerve damage, blindness, kidney failure, and heart failure. Statistics says, 382 million people worldwide have diabetes which will double to 592 million by 2035.**

**The diagnosis of diabetes can be made using a variety of conventional techniques based on macroscopic level and microscopic level examinations. However, because of the intricate interplay of numerous factors, early diabetes prediction is a difficult assignment for medical professionals. Data science methodologies have the potential to advance other scientific disciplines. To assist in making predictions based on medical data is one such endeavor. Data science has an emerging topic called machine learning that studies how machines learn from experience. The goal of this project is to create a system that can accurately forecast a patient's risk of developing diabetes by applying four different supervised machine learning techniques, and they are: Support vector Machine (SVM), Decision Tree, Random Forest and AdaBoost classifier.**

***Index Terms*—Classification, Normalization, data tuning, SVM, Random Forest, AdaBoost**

## I. Introduction

Many developing countries like Bangladesh diabetes is a very common disease. For instance, a large proportion of Bangladeshis suffer from the negative effects of diabetes, which are predicted to increase by 2025. It is a disease which leads people towards disability or death. According to a recent study by the International Diabetes Federation, the incidence of both diagnosed and undiagnosed cases is roughly similar among Bangladeshi nationals [1]. On many nations, the rising number of diabetic patients is expected to have a significant social and economic impact [2].

At the same time, with the growth of diabetic patient number the female diabetic patient is increasing at an alarming rate. Almost every women have the test of motherhood. During that critical time women experience hormonal change and sometime faces many health complexity. Moreover, women are physically different from men, that is why, the reasons of diabetes in women is slightly different from man.

One big issue is most of the time it is not possible to diagnosis diabetes on time. But for diabetes patient it is very important to maintain their daily diet because high level of sugar in blood and urine can create others medical emergencies. That is why, it is very important to detect this disease on time.

Classification is one of the most important supervised machine learning approaches to make decision for many real world problem. In this project, we have classified the data as diabetic or non-diabetic using four classification technique **Support vector Machine (SVM), Decision Tree, Random Forrest and AdaBoost classifier and have obtained accuracy of 73%, 76% (after tuning),77.6% (after tuning) and 100% respectively.**

## II. Literature Review

Results from related research that analyzed various healthcare datasets and made predictions using a variety of approaches and strategies are presented. Machine learning is a development of human-made brainpower that discovers connections between hubs without before preparing them in monasteries. [6] Researchers have created and used a variety of prediction models utilizing different data mining techniques, machine learning algorithms, or even a mix of these approaches.

kumar et al. [7] developed a method for the analysis of diabetic data using Hadoop and the Map Reduce approach. This approach is able to forecast the risk factors and type of diabetes. This solution is cost-effective for any healthcare business and is built on Hadoop . Sajida Perveen et al. [8] proposed AdaBoost technology. A diabetic patient should be arranged using an AdaBoost gathering model rather than

packing and J48. The creator is excited by the spread of diabetes across the world, and as a result, the anticipation and treatment of diabetes mellitus are gaining significance in the local medical community [9].

Matching the features of the data to be learnt with those of current methods is a step in selecting a machine learning algorithm. The following section discusses different machine learning algorithm taxonomy have been used in this research.

## III. Preliminary Concepts:

In this project we are going to use four machine learning algorithms three supervised learning algorithms and one un-supervised learning algorithm. Those algorithms are:

- Support vector Machine ( SVM).
- Decision Tree.
- Random Forrest.
- Adaboost classifier.

Brief details of those algorithms will be discussed on this point

### *A. Support Vector Machine(SVM):*

One of the most well-liked supervised learning algorithms, SVM, is used to solve Classification and Regression problems. The SVM algorithm's objective is to establish the best line or decision boundary that can divide n-dimensional space into classes, allowing us to quickly classify fresh data points in the future. A hyperplane is the name given to this optimal decision boundary. It selects the extreme vectors or points that aid in the creation of the hyperplane. Support vectors, which are used to represent these extreme instances, form the basis for the SVM method.

SVM can be of two types:

- **Linear SVM:** When a dataset can be divided into two groups using just one straight line, it is said to be linearly separable, and a classifier known as a Linear SVM classifier is used to separate the data into these two categories.
- **Non-linear SVM:** This SVM is used for non-linearly separated data, which implies that if a dataset cannot be categorized using a straight line, it is non-linear data, and the classifier employed is a non-linear SVM classifier.

### *B. Decision Tree:*

A supervised learning method called a decision tree may be used to solve classification and regression issues, but it is often favored for doing so. It is a tree-structured classifier, where internal nodes stand in for a dataset's characteristics, branches for the decision-making process, and each leaf node for the classification result. The Decision Node and Leaf Node are the two nodes of a decision tree. While Leaf nodes are the results of decisions and do not have any more branches, Decision nodes are used to create decisions and have numerous branches. The decisions or the test are performed on the basis of features of the given dataset.

It is known as a decision tree because, like a tree, it begins with the root node and grows on subsequent branches to form a structure like a tree. The CART algorithm, which stands for Classification and Regression Tree algorithm, is used to construct a tree. A decision tree only poses a question and divides the tree into subtrees according to the response (Yes/No) [5]. Fig 1 explains the general structure of a decision tree.

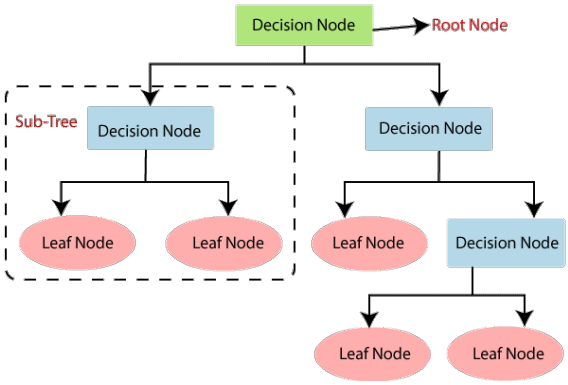


Fig. 1. general structure of a decision tree.

In a decision tree, the algorithm begins at the root node and works its way up to forecast the class of the provided dataset. This algorithm follows the branch and jumps to the following node by comparing the values of the root attribute with those of the record (real dataset) attribute. The algorithm verifies the attribute value with the other sub-nodes once again for the following node before continuing. It keeps doing this until it reaches the tree's leaf node [5].

The complete process can be better understood using the below algorithm:

- **Step-1:** S advises starting the tree from the root node, which has the whole dataset.
- **Step-2:** Utilize Attribute Selection Measure to identify the dataset's top attribute (ASM).
- **Step-3:** Subsets of the S that include potential values for the best qualities should be created.
- **Step-4:** Create the best attribute-containing decision tree node.
- **Step-5:** Utilizing the subsets of the dataset generated in step 3, iteratively design new decision trees. Continue down this path until you reach a point when you can no longer categorize the nodes and you refer to the last node as a leaf node.

### *C. Random Forest:*

Popular machine learning algorithm Random Forest, a supervised learning methodology, which may be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning [3]. As the name implies, is a classifier that uses a number of decision trees on different subsets of the provided dataset and averages them to increase the dataset's prediction accuracy. Higher accuracy and overfitting are prevented by the larger number of trees in the forest. Fig 2 explains the work flow of the Random Forest algorithm. Some decision trees may predict the proper

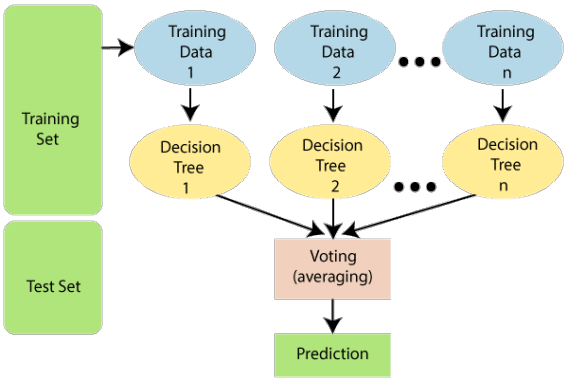


Fig. 2. Work flow of the Random Forest algorithm.

output, while others may not, since the random forest mixes numerous trees to forecast the class of the dataset. But when all the trees are combined, they forecast the right result [3]. Consequently, the following two presumptions for an improved Random forest classifier:

- For the dataset's feature variable to predict true outcomes rather than a speculated result, there should be some real values in the dataset.
- Each tree's predictions must have extremely low correlations.

The Working process can be explained in the below steps and diagram:

- **Step-1:** Pick K data points at random from the training set.
- **Step-2:** Create the decision trees linked to the chosen data points.
- **Step-3:** For any decision trees you intend to construct, choose N.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** Find the forecasts for each decision tree for new data points, and then allocate the new data points to the category that receives the majority of votes.

### *D. AdaBoost:*

First of all, AdaBoost is short for Adaptive Boosting, the first really successful boosting algorithm developed for binary classification. Also, it is the best starting point for understanding boosting. Moreover, modern boosting methods build on AdaBoost, most notably stochastic gradient boosting machines. Generally, AdaBoost is used with short decision trees. Further, the first tree is created, the performance of the tree on each training instance is used. Also, we use it to weight how much attention the next tree. Thus, it is created should pay attention to each training instance. Hence, training data that is hard to predict is given more weight. Although, whereas easy to predict instances are given less weight [4].

AdaBoost Model from Data:

Each instance in the training dataset is weighted. The initial weight is set to:

$$weight(xi) = 1/n \tag{1}$$

| Row No. | Gender | Age | Income | Illness | Sample Weights |
|---|---|---|---|---|---|
| 1 | Male | 41 | 40000 | Yes | 1/5 |
| 2 | Male | 54 | 30000 | No | 1/5 |
| 3 | Female | 42 | 25000 | No | 1/5 |
| 4 | Female | 40 | 60000 | Yes | 1/5 |
| 5 | Male | 46 | 50000 | Yes | 1/5 |

Fig. 3. Actual representation of AdaBoost dataset.

Where xi is the i'th training instance and n is the number of training instances.

A weak classifier is prepared on the training data using the weighted samples. Only binary classification problems are supported. So each decision stump makes one decision on one input variable. And outputs a +1.0 or -1.0 value for the first or second class value. The misclassification rate is calculated for the trained model. Traditionally, this is calculated as:

$$error = \frac{correct–N}{N} \tag{2}$$

Where error is the misclassification rate. While correct is the number of training instance predicted by the model. And N is the total number of training instances if the model predicted 78 of 100 training instances the error. This is modified to use the weighting of the training instances:

$$error = \frac{\sum_{i=1}^{n}(w_{\mathrm{i}} * terror_{\mathrm{i}})}{\sum_{i=1}^{n} w_{\mathrm{i}}} \tag{3}$$

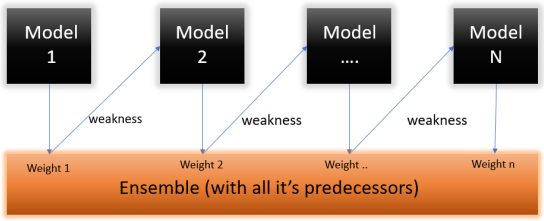


Fig. 4. workflow of AdaBoost.

Which is the weighted sum of the misclassification rate. where $w_{\mathrm{i}}$ is the weight for training instance i,

terror is the prediction error for training instance i. Also, which is 1 if misclassified and 0 if correctly classified. if we had 3 training instances with the weights 0.01, 0.5 and 0.2. The predicted values were -1, -1 and -1, and the actual output variables in the instances were -1, 1 and -1, then the terrors would be 0, 1, and 0. The misclassification rate would be calculated as:

$$error = \frac{(0.01 * 0 + 0.5 * 1 + 0.2 * 0)}{(0.01 + 0.5 + 0.2)} = 0.704 \tag{4}$$

A stage value is calculated for the trained model. As it provides a weighting for any predictions that the model makes. The stage value for a trained model is calculated as follows:

$$stage = \ln(\frac{1 - error}{error}) \tag{5}$$

```
In [120]: import matplotlib.pyplot as plt
          %matplotlib inline
          diabetes_dataset['Outcome'].hist(grid=False, legend=True)
          plt.show()
```

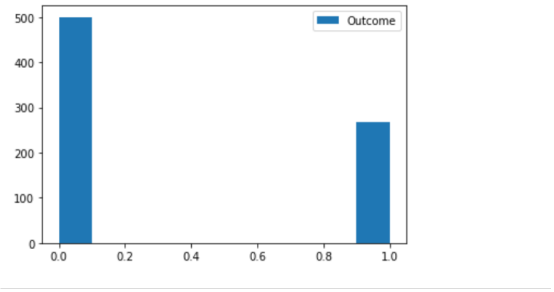


Fig. 5. Label count in dataset.

Where stage is the stage value used to weight predictions from the model. Also, ln() is the natural logarithm and error is the misclassification error for the model. The effect of the stage weight is that more accurate models have more weight. The training weights are updated giving more weight to predicted instances. And less weight to predicted instances.

AdaBoost Ensemble:

Basically, weak models are added sequentially, trained using the weighted training data. Generally, the process continues until a pre-set number of weak learners have been created. Once completed, you are left with a pool of weak learners each with a stage value. Predictions are made by calculating the weighted average of the weak classifiers. For a new input instance, each weak learner calculates a predicted value as either +1.0 or -1.0. The predicted values are weighted by each weak learners stage value. The prediction for the ensemble model is taken as a sum of the weighted predictions. If the sum is positive, then the first class is predicted, if negative the second class is predicted.

lists of some heuristics for best preparing your data for AdaBoost:

- **Quality Data:** Because of the ensemble method attempt to correct misclassifications in the training data. Also, you need to be careful that the training data is high-quality.
- **Outliers:** Generally, outliers will force the ensemble down the rabbit hole of work. Although, it is so hard to correct for cases that are unrealistic. These could be removed from the training dataset.
- **Noisy Data:** Basically, noisy data, specifical noise in the output variable can be problematic. But if possible, attempt to isolate and clean these from your training dataset.

So, this was all about AdaBoost Algorithm in Machine Learning.

## IV. Dataset

For this project we have used a public dataset from National Institute of Diabetes and Digestive and Kidney Diseases. The dataset contains 768 instances of female patient with 8 numeric-valued attributes like, Number of times pregnant, Plasma glucose concentration a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-Hour serum insulin (mu U/ml) and Age (years) etc..

The dataset is available in http://surl.li/cvstx. Fig 5 showes the label count in dataset.

## V. Methodology

First, we standardize our data and then split it into a training set and a test set where the test size was %. Then we implemented the Support Vector Machine Algorithm (SVM) to predict the diabetes condition of our patient. Our SVM kernel parameter was selected as “linear”. We have used Scikit Learn library to train the model. Then we tried some tree-based models to predict the diabetes condition. Here we used Decision Tree, Random Forest, and AdaBoost Algorithm.

At first, we created a basic decision model without tuning any hyperparameter. In the basic decision tree model, the criterion was selected “gini” and other parameters like max_depth, and min_sample_leaf were selected to default. Then we used GridSearchCV to find the best estimator and best parameters. Here ‘criterion’: ’entropy’, ’max_depth’: 5, ’min_samples_leaf’: 0.08 are the best parameters we get after tuning the hyperparameters.Then we train our model using the best estimator decision tree model.

Second, we used an ensemble model named random forest. Like before we created a base random forest model without hyperparameter tuning. Then we used GridSearchCV to do hyperparameter tuning and find the best hyperparameters and best estimator model. Here ”criterion’: ’gini’, ’max_features’: ’auto’, ’min_samples_leaf’: 2, ’n_estimators’: 250 are the best parameters we get after tuning the hyperparameters.

Finally, we used a boosting model named AdaBoost for estimating the result. Here we used the best estimated tuned decision tree model as the base estimator of the AdaBoost without hyperparameter tuning the other parameters of the AdaBoost model. After that, we tuned the hyperparameter of our AdaBoost model using GridSearchCV. ’$n_estimators' should be 200 in the hyperparameter - tuned AdaBoost model$.

## VI. Result

TABLE I
Accuracy of different model

| *ML Model* | *Accuracy* |
|---|---|
| SVM | 73.37% |
| Decision tree | 76% |
| Random forest | 77.6% |
| AdaBoost | 100% |

In this section we have discussed about the output obtained by our research **SVM result:**From the SVM base model our training accu racy score is 0.7838 and the test accuracy score is 0.733766

**Decision Tree result:** The accuracy score of the training data for the basic decision tree model is 0.71354 After hyperparameter tuning the training accuracy score becomes 0.760

**Random Forest result:** The accuracy score of the training data for the basic random forest The ensemble model is 0.77083. After hyperparameter tuning, the training accuracy score becomes 0.7760416

**AdaBoost result:** The accuracy score of the training data for the basic AdaBoost model is 0.765625 After hyperparameter tuning the training accuracy score becomes 1.0

## VII. Conclusion

After comparing the accuracy of each classification model, it can be said that AdaBoost is the most accurate approach for our dataset, with an accuracy of 100%. The performance was the worst for Decision Tree. As a result, an early diabetes diagnosis prediction system based on the AdaBoost model has a lot of potential. Careful data should be integrated as a vital component for future examination of these outcomes. To assess our present findings, we thus intend to test these classification models using other datasets to get more reliability of our research.

## References


[1] P. M. S. Sai, G. Anuradha, and V. P. Kumar, "Survey on type 2 diabetes prediction using machine learning," in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), 2020.

[2] P. K. Dhillon et al., "Status of epidemiology in the WHO South-East Asia region: burden of disease, determinants of health and epidemiological research, workforce and training capacity," Int. J. Epidemiol., vol. 41, no. 1, pp. 848–860, 2012.

[3] "Random forest algorithm," www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/machine-learning-random-forest-algorithm. [Accessed: 31-Aug-2022].

[4] A. Saini, "AdaBoost Algorithm - A complete guide for beginners," Analytics Vidhya, 15-Sep-2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/. [Accessed: 31-Aug-2022].

[5] "Decision tree classification algorithm," www.javatpoint.com. [Online]. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm. [Accessed: 31-Aug-2022].

[6] "Diabetes: Asia's 'silent killer'", November 14, 2013". Available at: www.bbc.com/news/world-asia-24740288.

[7] N. M. S. kumar, T. Eswari, P. Sampath, and S. Lavanya, "Predictive methodology for diabetic data analysis in Big Data," Procedia Computer Science, vol. 50, pp. 203–208, 2015.

[8] Perveen S, Shahbaz M, Keshavjee K, Guergachi A. Metabolic syndrome and development of diabetes mellitus: predictive modeling based on-machine learning techniques, IEEE Access. IEEE. 2019;7: 1365–75. https://doi.org/10.1109/ACCESS.2018.2884249.

[9] Perveen S, et al. Performance analysis of data mining classification techniques to predict diabetes. Procedia Computer Science. 2016;82:115–21. https://doi.org/10.1016/j.procs.2016.04.016 Elsevier Masson SAS