INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 5)
Available online at: www.ijariit.com

Predicting Diabetes using Machine Learning Technique

A. K. Aravind Kumar
aravindmcs1@gmail.com
Indian School of Business, Bangalore, Karnataka

ABSTRACT

Diabetes is a chronic illness with the potential to induce a worldwide health care crisis. As per the International Diabetes Federation, 382 million people exist with diabetes in the world currently. By 2035, the number is expected to be doubled as 592 million. Diabetes mellitus or diabetes is a disease generally induced due to the grown level of glucose in the blood. Physical and chemical tests are different traditional methods for diagnosing diabetes. However, diabetes prediction in advance is pretty challenging for medical practitioners due to complicated interdependence on multiple factors as diabetes influences individual organs such as eye, heart, kidney, eye, nerves, foot, etc. Data science techniques have the potential to help the medical field by answering some of the general questions. One such task is to facilitate predictions on medical data. Machine learning is the most useful technology for the medical field in data science. Machine learning is helpful because of the way machines learn from experience. This project aims to propose a valuable technique for earlier detection of the diabetes disease for a patient with higher efficiency by combining the outcomes of different machine learning techniques, the supervised machine learning methods including Decision Tree, Logistic Regression, Random Forest, Neural Network, XGBoost, and Support Vector Machine.

Keywords: *Diabetes, Machine Learning, Decision Tree, Logistic Regression, Random Forest, Neural Network, XGBoost, Support Vector Machine*

1. INTRODUCTION

1.1 What Is Diabetes Mellitus?

Diabetes mellitus (DM) is known as diabetes, is one of the most life-threatening diseases in the world. It is not only an ailment but also a producer of various kinds of diseases like heart stock, blindness, kidney diseases, etc. The traditional identifying method is that patients are required to visit a diagnostic centre, consult their physician, and wait for days to get their reports. Moreover, they have to waste their time and money to get diagnosed. These methods can be automated by using Diabetes recognition Software based on Machine Learning. Diabetes Mellitus (DM) is a condition that restricts the body from correctly using the energy from the meal eaten. Diabetes happens in one of the following terms:

An organ behind the stomach, which is called the pancreas, produces insufficient insulin or no insulin. Insulin is a hormone that occurs naturally, produced by the pancreas' beta cells, which aids the body use sugar for energy. As earlier discussed, the pancreas produces insulin, but the insulin produced doesn't work as it should. This state is called insulin resistance.

The body consists of millions of cells. The cells need food in a particular form to make sufficient energy. When we eat or drink, our food is broken down into pure sugar, which is called glucose. Glucose will be converted to energy and circulated to our body muscles to perform our daily activities. Diabetes is two types: Type 1 and Type 2:

Type 1 diabetes occurs because the pancreas may be damaged or produces insufficient insulin or no insulin. Insulin is a hormone that occurs naturally, produced by the pancreas' beta cells, which aids the body use sugar for energy. So sugar can't go into the human body's cells for usage as energy. Patients with Type 1 diabetes usually suggested using insulin injections to manage their blood glucose.

Type 2 diabetes, generally known as adult-onset diabetes, in this case, the pancreas makes insulin, but it doesn't generate enough or doesn't work correctly. Type 2 diabetes usually can be regulated with a good diet, weight management, and proper exercise. However, the treatment may include oral glucose or insulin injections or a combination of both.

1.2 Machine Learning

Machine Learning is the course of study that provides computers with the intelligence to learn without doing much programming.

ML is one of the most persuasive techniques that one would have ever occurred across. Machine Learning (ML) can be explained as automating and improving computers' learning processes based on their experiences without being coding the program. The process starts with feeding useful quality data and then training our computers by building machine learning models using the data and different algorithms.

Fig. 1: Basic Difference in ML and Traditional Programming.

Traditional Programming: We feed in DATA (Input) + PROGRAM (logic), run it on the machine, and get output.
Machine Learning: We feed in DATA (Input) + Output, run it on the machine during training, and the computer creates its program (logic), which can be testing in the validation process.

1.3 Supervised Learning

Supervised learning is generally called when the predictive model is getting learned on a labelled dataset. The labelled data is one that has input parameters as well as output parameters. In this kind of learning, both training and testing datasets are labelled, as shown in the image below.

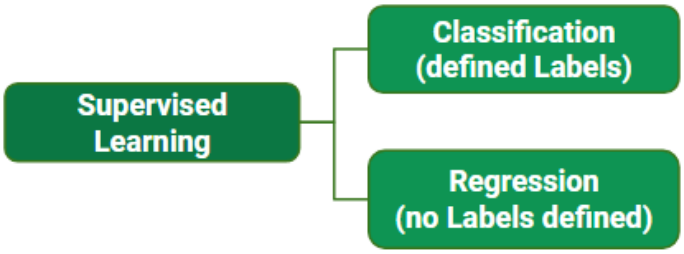

Fig. 2: Supervised Learning and types of Supervised Learning.

1.3.1 Types of Supervised Learning

a) **Classification:** It is one of the Supervised Learning applications where output is holding specified labels (discrete value). For example, if the output variable has defined labels that are 0 or 1, 1 means the customer will purchase (for example), and 0 means that the customer won't buy. The aim here is to predict discrete values relating to an appropriate class and evaluate it based on accuracy. It can be either binary classification or multi-class classification. In the case of binary classification, the fitted model predicts either 0 or 1; yes or no, whereas in the situation of multi-class classification, the fit model predicts more than one class.

b) **Regression:** It is another example of Supervised Learning, where output has continuous value. For instance, Output – Wind Speed (target variable in the case) does not hold any discrete value, but it is continuous in some value range. The aim here is to predict the target variable value as close to the actual output value as our model can. Then, in the evaluation process, the error value will be calculated to draw intuitions on data—the less the error, the higher the precision of the regression model.

1.3.2 Below are some examples of Supervised Learning Algorithms

Gaussian Naive Bayes, Random Forest, Nearest Neighbor, Decision Trees, Linear Regression, Support Vector Machine (SVM)

1.4 Unsupervised Learning

In Unsupervised Learning, a training model has only input parameter values. Generally, in unsupervised learning, the model itself has to find the way that model can learn from the data. For example, consider the subscription data to the particular mall; once subscribed, the customers will be provided with a membership card, so the mall has to process the customer information for each purchase. Using this membership customer data and unsupervised learning techniques, the mall can easily group clients based on the parameters.

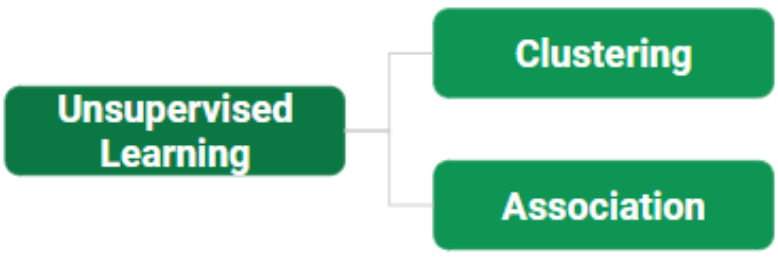

Fig. 3: Unsupervised Learning and types of Unsupervised Learning

1.4.1 Types of Unsupervised Learning

a) **Clustering:** this technique is used to segment the data based on patterns or specific condition in the data that the machine-learning model finds.
b) **Association:** this technique works on relations between parameters in the massive data and applies rules to fit the right model. Generally, it is called a rule-based Machine Learning technique.

1.4.2 Some examples of Unsupervised Learning Algorithms

K-Means Clustering, Hierarchical Clustering, and Clustering using Hierarchies.

1.5 Semi-supervised Learning

This technique is useful when the data provided is with some labelled and other portion is un-labelled. Generally, we can use unsupervised techniques to predict missing labels first and then give complete data to supervised methods to create the right model. Typically, this kind of semi-supervised learning technique will be useful for image datasets, where images are not labelled.

Fig. 4: Semi-supervised Learning or Reinforcement flow chart.

1.6 Reinforcement Learning

This technique is beneficial where regular feedback is required for model learning. In this method, the model keeps on learning with reward feedback, so it increases its performance continuously. These algorithms are not used to solve the typical business problems; these algorithms are for specific issues like AlphaGO (bot competitions with the human), Google driving car, etc. Each time with a new data feed, the model learns with feedback and adds to its knowledge as the training process. So the model will be better trained with more it determines with feedback.

Example of Semi-supervised Learning Algorithms:

Deep Adversarial Networks, Q-Learning, and Temporal Difference (TD).

2. LITERATURE REVIEW

Sajida et al. [1] in this paper author explained the role of Adaboost and Bagging Ensemble techniques [2] using J48 decision tree as the basis for classifying the patients as diabetic or non-diabetic, based on diabetes chance factors. The author's experiment results prove that Adaboost machine learning ensemble technique performs well in terms of prediction comparatively bagging as well as a J48 decision tree.

Orabi et al. [3] in this paper, the author, discussed a new system of a method for diabetes prediction, which is incredibly helpful for who is suffering from diabetes at a particular age group. The suggested approach is composed based on the concept of machine learning by using a decision tree. Achieved results were adequate as the proposed method works well in predicting diabetes with at a particular age, with greater accuracy using Decision tree[4], [5].

Pradhan et al. [6] applied Genetic programming (GP) for the training and validation to predict diabetes on the Diabetes data set, which is taken from the UCI repository. Genetic Programming [7],[8] is very promising in prediction and gives optimal precision as compared to other executed techniques as per the experimental results. GP technique takes less time for execution and produces accurate results in case of classifier generation.

Rashid et al. [9] created a prediction model with two combinations of modules to predict diabetes as diabetic or non-diabetic. The first module is ANN (Artificial Neural Network), and the second module is Decision Tree. And it turned out to be FBS (Fasting Blood Sugar) is the critical factor to predict signs of diabetes.

Nongyao et al. [11] utilized an algorithm, which divides the risk of diabetes mellitus. To achieve the goal author has applied below machine learning techniques, those are Randomforest, ANN, Logistic Regression and Naive Bayes. Bagging and Boosting techniques are used to improve the performance and accuracy of the designed model. The experimental results reviled that the Random Forest algorithm gives optimum results among all the algorithms employed.

3. PROPOSED SOLUTION

Classification is one of the essential decision-making techniques in many real-world problems. In this work, the primary purpose is to classify the data as diabetic or non-diabetic and improve classification accuracy. For many classification problems, the higher

number of samples chosen, but it doesn't lead to higher classification accuracy. In many cases, the performance of the algorithm is high in the context of speed, but the accuracy of data classification is low. The main goal of our model is to achieve high efficiency. Classification accuracy can be increased if we use a bit high volume of the data for training and a few data for validation. This survey has examined various classification techniques for the classification of diabetic and non-diabetic data. Thus, it is observed that methods like Support Vector Machine(SVM), Logistic Regression(LR), and Artificial Neural Network(ANN) are most suitable for implementing the Diabetes prediction system.

4. MACHINE LEARNING ALGORITHMS

4.1 Decision Tree

Decision Tree algorithm belongs to the house of supervised learning algorithms. Decision Tree classification is a bit different than the other supervised learning algorithms; the decision tree algorithm can be applied for solving regression and classification problems too. The data is continuously divided according to a particular parameter. The tree can be distinguished by two entities, specifically decision nodes and edges. The edges are the decisions or final outcomes. And the decision nodes are where the data is split. A decision tree is a hierarchical tree-like graph with nodes and edges describing the place where we pick a characteristic and ask a problem; leaves represent the answers the to the question, and the edges represent the exact output or class label. They are used in non-linear decision making with simple linear decision surface. Decision trees classify the attributes in the problem by shrinking them from the root to some leaf node in the entire tree, with the edge node providing the classification to the problem. Each node in the whole tree acts as a test case for some attribute, and each edge sliding from that node corresponds to one of the possible answers to the test case. This process is recursive and is repeated for every subtree rooted at the new nodes.

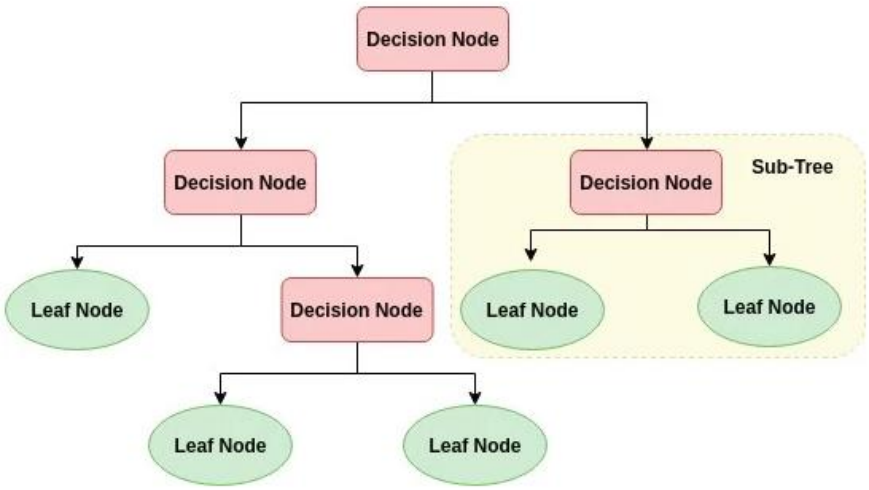

Fig. 5: Sample decision tree with nodes

4.2 Logistic Regression

Logistic regression is a classic classification technique, used when the value of the target variable is categorical. Logistic regression is usually used if the data in problem has binary output, so when it belongs to one class or another, i.e. 0 or 1. Logistic regression comes under the umbrella of supervised learning classification algorithms mainly used to predict the probability of a target variable. Logistic regression does an equation as the representation, very much like a linear regression equation. Input values (x) as shown in the below formula, are joined linearly using weights or coefficient values (β0, β1, etc.) to predict an output variable (y). A key differentiation from linear regression is that the output variable predicted as a binary value (0 or 1) rather than a continues numeric value.

Logistic regression equation:

$$y = \frac{e^{(\beta 0 + \beta 1 * x)}}{(1 + e^{(\beta 0 + \beta 1 * x)})}$$

In the above formula, y is the targeted output, where β0 is the bias or intercept term, and β1 is the coefficient or weight for the particular input value (x). Each column in the input data has an associated β coefficient or weight (a constant real value) that need to be learned from the training data. The realistic representation of the model that would be stored in memory or the file is the coefficients in the equation (β's).

4.3 Random Forest

The random forest is one of the popular algorithms among the supervised learning algorithms. This technique works in a way that it randomly creates and consolidates various decision trees into one "forest." The aim is not to rely on a single learning model, but rather a combination of decision models to improve accuracy. The principal difference between this method and the standard decision tree method is that the root nodes feature splitting nodes are generated randomly.

The Random forest comes under the bucket of supervised learning methods. The "forest" it builds with multiple decision trees, generally trained with the "bagging" technique. The common idea of the bagging method is that a mixture of learning models increases the overall result.

One significant advantage of random forest is that it can be used as classification problems and also regression, which create the majority of modern supervised learning systems. Below image shows a random forest with two trees:

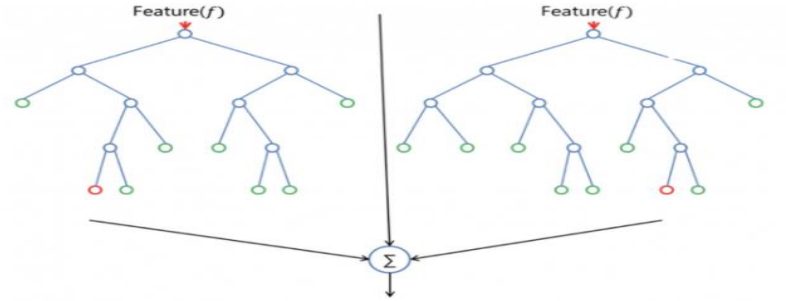

Fig. 6: Sample Random forest

The random forest has approximately the same parameters as a bagging classifier or a decision tree. Fortunately, there's no need to join a decision tree with a bagging classifier because of the ease of use of the classifier-class of random forest. With a random forest, using the algorithm's regressor also can solve regression tasks.

Usually, random forest attaches additional randomness to the model while building the trees. Instead of exploring for the most critical feature while splitting a node, it explores for the best feature among a random subset of existing features. This method results in a wide variation that generally appears in a better model.

Hence, in the random forest method, only a random subset of the characteristics is brought into consideration by the algorithm for cutting a node. Thresholds that even make trees more random by additionally using random thresholds for each feature rather than exploring for the best feasible thresholds (like a regular decision tree does).

4.4 Neural Network

An Artificial Neural Network (ANN) learning algorithm or neural network is a bit massive computational learning system that works with a network of functions to interpret and translate a data input of one form into the desired output. The theory of the artificial neural network was motivated by human biology and mimic the way the neurons of the human's brain works to understand inputs from human senses.

Artificial Neural Networks (ANN) are just one of the many tools and methods used in machine learning techniques. The artificial neural network itself may be used as a part in many complex machine-learning algorithms to process complicated data inputs into a space that computers can understand.

Artificial Neural networks are being used in various real-life problems, including spam email filtering, speech and image recognition, medical diagnosis and finance.

Machine learning algorithms that use neural networks are not required to be programmed with special rules. Instead, the neural net learning algorithm learns from input data by processing many labelled examples that are provided during the training process. And it continues the process of learning by using the answer key to learn what characteristics of the input are required to construct the correct output. Once the required number of examples (data inputs) has been processed, the neural network can start to process new, unseen inputs, and it predicts or produces accurate results successfully. If the model processes more input data and a variety of information, then, the more precise the results will be made because the program learns with experience.

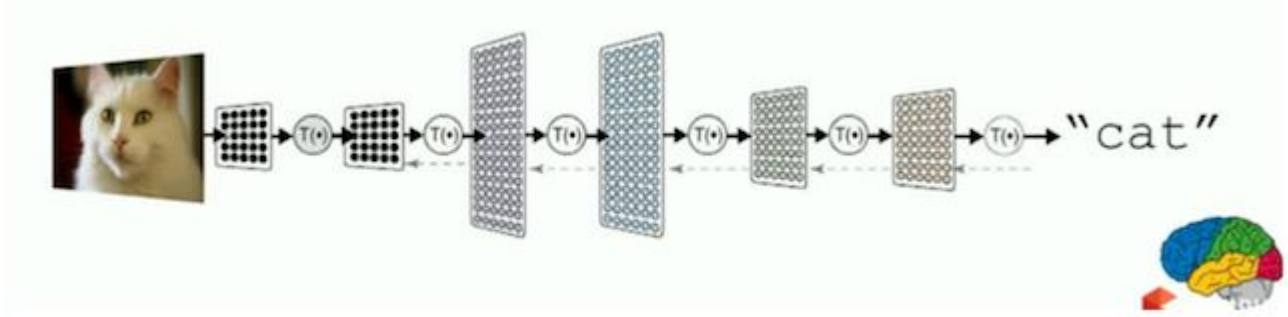

Fig. 7: Image processing with Neural Network

This theory can best be understood with an illustration. Imagine the problem is to figure out to determine whether or not a picture contains a cat. It is straightforward for a human to figure out. But it is much more challenging to train a computer to recognize a cat in a picture using classical techniques. Considering the various possibilities of how a cat may look in an image, writing program to consider for every scenario is almost impossible. But using machine learning, and especially using neural networks, the program can use a generalized method to understand the content of the image. Using different layers of functions to decompose the picture into data points and information that a computer can handle, the neural network can begin to recognize trends that exist across the many, many data inputs that it processes and classify images by their identities.

Once the model processes many training data of cat images, the algorithm determines what elements, and their corresponding relationships, in an image are essential to consider when deciding whether a cat is present in the image or not. When processing a new image, the ANN matches the data points regarding the new image to its model, which is built based on previous learning on the provided data. It then applies some simple statistics to determine whether the picture contains a cat or not based on how closely it meets the model.

In this case, the layers of functions between the input and the output are built as a neural network. In tradition, the neural network is somewhat more complex than the image above shows. The following picture captures the interaction between layers, but in real problems, there are many variations of the relationships between nodes, or neurons.

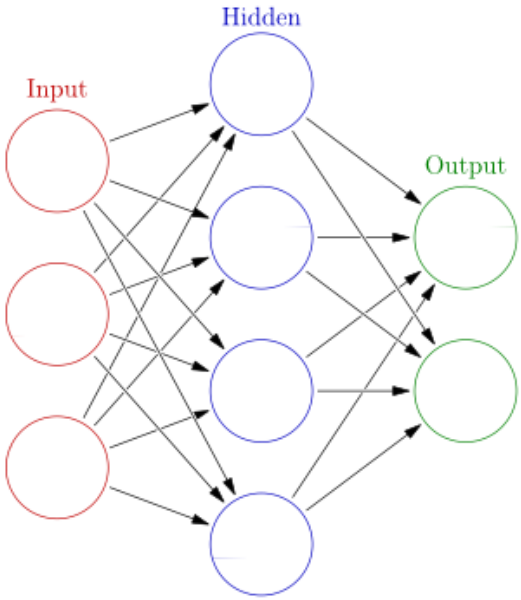

Fig. 8: Input, Output and Hidden layers structure in Neural Network

4.5 XGBoost

XGBoost is a traditional decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting as a framework. In the case of unstructured data prediction problems, artificial neural networks tend to exceed all other algorithms or frameworks. But, in the case of small-to-medium structured/tabular data, decision tree-based algorithms are admitted best-in-class.

4.5.1 Model Features: Three primary Gradient boosting forms are supported:

- Gradient Boosting algorithm, which is also called Gradient boosting machine that includes the learning rate.
- Stochastic Gradient Boosting considers sub-sampling at the row level, column level and column per split levels.
- Regularized Gradient Boosting with both L1 and L2 regularization.

4.5.2 System Features: The library which provides an arrangement for use in a range of computing contexts, not least:

- Parallelization of tree development using all of your CPU cores while training.
- Distributed Computing for training extensive models using a cluster of machines.
- Out-of-Core Computing for massive datasets that cannot be handled by computer memory.
- Cache Optimization of data structures and data used by the model to take great advantage of the use of computer hardware.

4.5.3 Algorithm Features: The implementation of the model was engineered to make the best use of CPU time and memory resources. A design goal was to increase the efficiency of available resources to train the model. Some key model implementation features include:

- Sparse Aware implementation which handles missing data values automatically.
- Block Structure to support the parallel process of tree construction.
- Continues training helps to further boost to fitted model on new data to increase model accuracy.
- XGBoost is free and open-source software available for use.

4.5.4 Why Use XGBoost?

Below are the two reasons to use XGBoost; usually, these two are goals any project:

- Execution Speed.
- Model Performance.

4.6 Support Vector Machine (SVM)

An SVM model is a design of different classes in a hyperplane in multidimensional space. The hyperplane will be formed iteratively by SVM so that the error can be minimized. The goal of SVM is to split the datasets into classes to obtain a Maximum Marginal Hyperplane (MMH).

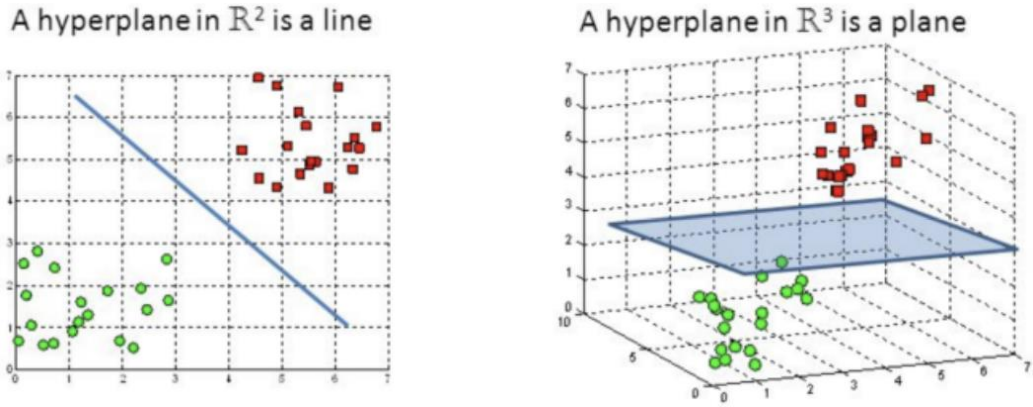

Fig. 9: Hyperplane and Support Vector Machine 2D and 3D images

The followings points are essential concepts in SVM −

- Support Vectors – Data points that are nearest to the hyperplane are called support vectors. Separating line will be established with the aid of these data points.
- Hyperplane − As shown in the above picture, it is a decision plane or space which is split between a set of objects having different classes.
- Margin − It is marked as the gap between two lines on the closet data points of different groups. It can be measured as the perpendicular distance from the line to the support vectors. Large margin is known as a good and small is considered as a bad margin.

The main aim of SVM is to split the datasets into classes to find a maximum marginal hyperplane (MMH), and it will be achieved by following two steps −

- First, SVM will form hyperplanes iteratively that divides the classes in the best way.
- Then, it will determine the hyperplane that separates the classes accurately.

5. CONCLUSION

Machine learning has the incredible ability to revolutionize the diabetes risk prediction with the aid of advanced computational techniques and the availability of an extensive amount of epidemiological and genetic diabetes risk data. Detection of diabetes in its initial stages is the key to treatment. This work has outlined a machine learning approach to predict diabetes levels in the human body. The method may also help researchers to develop an accurate and helpful tool for medical practitioners to help them make more reliable decisions about the disease status.

6. REFERENCES

[1] Guergachi, A., Keshavjee, K., Perveen, S., Shahbaz, M., 2016. Performance and Analysis of Data Mining Classification Techniques to Predict Diabetes. The Procedia Computer Science 82, 115–121. Journal DOI:10.1016/j.procs.2016.04.016.

[2] Nai-Arun, N., Sittidech, P., 2014. Ensemble Learning Model for Diabetes Classification. Advanced Materials Research 931 - 932, 1427–1431. doi:10.4028/www.scientific.net/AMR.931-932.1iques 427.

[3] Kamal, Y.M., Rabah, T.M., Orabi, K.M., 2016. Predictive System for Diabetes Mellitus Disease, Industrial Conference on Data Mining. Springer. pp. 420–427.

[4] Rathee A., Priyam A., Srivastava S., Gupta R., Comparative Analysis of Decision Tree Classification Algorithms for Diabetes. IJCET Vol.3, 334–337. DOI:arXiv: ISSN 2277 - 4106 JUNE 2013.

[5] Kay J., Esposito F., Malerba D., Semeraro G., 1997. The comparative analysis of machine learning methods for pruning decision trees. IEEE. 476–491. DOI:10.1109/34.589207.

[6] G.R., Bamnote M.P., 2014. The Design of Classification for Detection of Diabetes Using Genetic Programming. AISC Advances in Intelligent Systems and Computing 1, 763–770. DOI:10.1007/978-3-319-11933-5.

[7] Sheta, A., Sharief A.A., 2014. Developing a Statistical Model to Detect Diabetes Using Multigene Genetic Programming. (IJARAI) 3, 54–59. DOI:10.14569/IJARAI.2014.031007.

[8] Dhobale, V., Pradhan, P.M.A., Bamnote, G.R., Tribhuvan, V., Jadhav, K., Chabukswar, 2012. A Genetic Programming method for Detection of Diabetes. (IJOCE) 2, 91–94.

[9] Abdullah, R.M., Tarik A., Rashid, S.M.A., Abstract, 2016. An Intelligent Approach for Diabetes Classification, Prediction and Description. Advances in Intelligent Systems and Computing 424, 323–335. DOI:10.1007/978-3-319-28031-8.

[10] Han, J., Rodriguez, J.C., Beheshti, M., 2008. Discovering decision tree-based diabetes prediction model, International Conference on Advanced Software Engineering and Its Applications, Springer. pp. 99–109.

[11] Nai-Arun, N., Moungmai, R., 2015. Comparison of Classifiers for the Diabetes Prediction. Procedia Computer Science 69, 132–142. DOI:10.1016/j.procs.2015.10.014.

