2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC)

Diabetic Prediction Using Machine Algorithm SVM and Decision Tree

Chaitanya Sonawane
Department of *Information Technology*
Pimpri Chinchwad College of Engineering
Pune, India
chaitanyasonawane2102@gmail.com

Kalpesh Somwanshi
Department of *Information Technology*
Pimpri Chinchwad College of Engineering
Pune, India
kalpeshksomwanshi910@gmail.com

Raj Patil
Department of *Information Technology*
Pimpri Chinchwad College of Engineering
Pune, India
patilraj1227@gmail.com

Roshani Raut
Department of *Information Technology*
Pimpri Chinchwad College of Engineering
Pune, India
roshani.raut@pccoepune.org

***Abstract*— Diabetes is a serious metabolic disorder that affects millions of people globally. Early detection and management of diabetes are essential to prevent severe complications. In recent years, machine learning algorithms have become increasingly popular in the medical field to predict the onset of diabetes. This study aims to predict the onset of diabetes using the support vector machine (SVM) and decision tree algorithms. The dataset used for this study is the Pima Indian diabetes dataset, which contains several features such as glucose, insulin, and body mass index. The problem statement is to determine which algorithm is more accurate in predicting diabetes. The methodology involves implementing the SVM and decision tree algorithms on the dataset and evaluating their performance using metrics such as accuracy, precision, and recall. The results of the study indicate that the SVM algorithm performs better than the decision tree algorithm, with an accuracy of 76.6% compared to 75%. This work concludes that the SVM algorithm is more accurate in predicting diabetes and can be a valuable tool for early detection and management of the disease. This study provides insight into the potential use of machine learning algorithms in the medical field and highlights the need for further research to improve the accuracy of diabetes prediction models.**

I. INTRODUCTION

The prevalence of diabetes, a chronic illness affecting numerous individuals globally, is quickly on the rise. One of the most challenging health concerns globally is diabetes mellitus [1]. According to the International Diabetes Federation, the number of people affected by diabetes globally is currently more than 285 million and is projected to increase to 380 million within the next 20 years [1]. Therefore, the development of a classifier for detecting diabetes with optimal cost and high accuracy is critical. This disease can lead to serious complications such as heart disease, kidney disease, blindness, blood vessel damage, and nerve damage. The condition arises when the body is incapable of generating the necessary insulin to control the sugar levels in the body. There are two types of diabetes: Type 1, in which the pancreas does not make enough insulin, and Type 2, in which the insulin produced is not effective in activating the cells. 90-95% of people are affected by Type 2 diabetes [2]. Diabetes has become an increasingly significant health issue globally, with prevalence among adults rising from 4.7% to 8.5% between 1980 and 2014 [3]. Second third world countries are particularly affected, and projections indicate that the number of people living with diabetes will continue to rise, with estimates suggesting 693 million people with diabetes by 2045 [4].

Early detection and accurate diagnosis of diabetes are crucial to managing the disease and preventing complications. However, the process of diabetes diagnosis can be challenging, and physicians have to analyze many factors before making a diagnosis, which can make their job difficult. In recent years, machine learning and data mining techniques have been explored as promising tools for automatic diabetes diagnosis. Such techniques can provide a cost-efficient, convenient, and accurate means of classification, thereby enabling early detection and diagnosis. Machine learning algorithms, including Support Vector Machine (SVM) [4], Genetic Algorithm (GA) [5], Decision Tree [6], and Neural Networks, are becoming popular in biomedical and bioinformatics for predicting various diseases, such as chronic kidney disease (CKD). SVM, a classification method for linear and non-linear systems, is efficient in memory usage and capable of handling high-dimensional input spaces [7]. It can be effective even with a small number of samples and offers various kernels for decision-making, including custom kernels. SVM has shown success in diagnosing critical diseases, including CKD, and can uncover hidden patterns from data.

The diabetic database of the Pima Indians, which is available at the machine learning laboratory of the University of California, Irvine, is a well-known dataset that is commonly used to evaluate the performance of data mining algorithms in classifying diabetes [3]. This study proposes the use of a Support Vector Machine (SVM), a powerful machine learning algorithm, as a classifier for diagnosing diabetes. SVM is specifically tailored to classify diabetes from high-dimensional medical datasets [7]. Experimental results from the study show that SVM can be used successfully to diagnose diabetes. SVM's performance was evaluated using

979-8-3503-2379-5/23/$31.00 ©2023 IEEE

various metrics, such as precision, recall, and accuracy. The study's results suggest that the proposed SVM classifier could be a useful tool for the early detection and prevention of diabetes.

This research paper aims to explore the effectiveness of SVMs in diabetes diagnosis. The study is based on an extensive experimental analysis of a diabetes dataset, evaluating the performance of SVMs in terms of accuracy, sensitivity, specificity, and other relevant metrics. The researchers will compare the performance of SVMs with other commonly used machine learning techniques in the area of diabetes diagnosis.

The organization of the rest of this paper can be summarized as follows: Section II has Related works for this paper, Section III contains Methodology, Section IV shows Results and Analysis and finally, Section V gives the conclusion of the paper.

II. Related Work/ Literature Survey

Diabetes is a chronic disease that affects millions of people worldwide, and its early detection is essential to prevent or delay the onset of its complications. Machine learning (ML) algorithms have been used to predict the risk of developing diabetes based on various features, such as age, body mass index (BMI), family history, and blood glucose levels.

1. "Diabetes Prediction using Machine Learning Techniques: A Review" [8] by Suresh Kumar, Shailendra Kumar Singh, and Shikha Gupta (2021) - This paper provides a comprehensive review of the literature related to diabetic prediction using ML techniques. The authors discuss various ML algorithms, including decision trees, support vector machines (SVMs), neural networks, and random forests, and their performance in predicting diabetes.

2. "Machine learning-based prediction models for diabetes: A systematic review " [9] by Haitham M. Yousef, Eman Almaliky, and Mohammed H. Alshammari (2021) - The authors conducted a systematic review of the literature related to ML-based diabetic prediction models. They identified 30 studies that used various ML algorithms, including logistic regression, SVMs, decision trees, and neural networks. The authors concluded that ML algorithms have the potential to accurately predict the risk of diabetes.

3. "Machine Learning Techniques for Diabetes Prediction: A Review " [10] by Rishabh Jain and Ankit Sharma (2020) - The authors reviewed various ML algorithms, including SVMs, decision trees, and random forests, for diabetic prediction. They also discussed the importance of feature selection in improving the performance of ML algorithms. The authors concluded that SVMs and random forests are the most effective algorithms for diabetic prediction.

A. Support Vector Machine

Support Vector Machine (SVM) [4] is a supervised machine learning algorithm that has been widely used for classification, regression, and outlier detection tasks. The algorithm is known for its ability to handle high-dimensional data efficiently and provide accurate predictions in various applications.

In a binary classification problem, SVM tries to find the hyperplane that maximizes the margin between the two classes [11]. This margin is defined as the distance between the hyperplane and the closest data points from each class. SVM can handle both linearly separable and non-linearly separable data by using a technique called the kernel trick. The kernel trick maps the data into a higher dimensional space where it becomes linearly separable. This approach can be used with different types of kernels to handle different types of data.

SVM has several advantages over other classification algorithms. It has a strong theoretical foundation, which makes it easier to understand and interpret [12]. SVM can handle high-dimensional data efficiently and is useful when the goal is to find a clear boundary between different classes. Additionally, SVM can be used for both binary and multi-class classification problems.

However, SVM can be sensitive to the choice of kernel and the regularization parameter [14]. The performance of SVM may suffer when the number of features is much larger than the number of training instances. Furthermore, SVM may not perform well on very large datasets.

In conclusion, SVM is a powerful machine learning algorithm that can be used for a variety of tasks, especially when the data is high-dimensional and the goal is to find a clear boundary between different classes. SVM has several advantages over other classification algorithms, including its ability to handle high-dimensional data efficiently and its strong theoretical foundation. However, SVM may be sensitive to the choice of kernel and regularization parameter, and its performance may suffer on very large datasets.

B. Decision Tree

Decision trees [6] are a popular and widely used machine learning and data mining model. They provide a graphical representation of a series of conditional rules or decisions that can be used to make predictions or decisions. Decision trees are easy to understand and interpret, which makes them a popular choice for a wide range of applications.

The decision tree model consists of nodes and branches. Each node represents a decision point based on a specific condition or feature, and each branch emanating from a node represents one possible outcome of that decision. This process continues until the terminal nodes are reached, which represent the outcome or prediction. Decision trees can be used for classification or regression problems, depending on the type of data being analyzed.

Classification trees are used when the target variable is categorical, while regression trees are used when the target variable is continuous. Decision trees can be used with both structured and unstructured data and are popular because of their ease of use and interpretability.

However, decision trees can be prone to overfitting if not properly pruned, and they may not perform as well as other models in some situations [10]. Therefore, researchers have proposed various techniques to overcome these limitations. For example, ensemble methods such as Random Forest and Gradient Boosting have been developed to improve the accuracy and robustness of decision trees.

The study compared the performance of decision trees, logistic regression, and artificial neural networks on a medical dataset. The study found that decision trees had the highest accuracy and outperformed the other two models in terms of interpretability and ease of use.

In conclusion, decision trees are a popular and widely used model in machine learning and data mining. While they have some limitations, researchers have developed various techniques to overcome these limitations, and decision trees continue to be useful tools for making predictions and decisions in a wide range of applications.

III. METHODOLOGY

The dataset [15], sourced from Kaggle, was originally provided by the National Institute of Diabetes and Digestive Diseases. It includes various features such as glucose level, insulin level, body mass index (BMI), and age.

- *Using SVM (Support Vector Machine)*

A. *Experimental Setup of SVM for Diabetes Prediction*

SVM is a machine learning algorithm used for classification tasks, including disease prediction. The experimental setup for SVM-based disease prediction typically involves the following steps:

1. Collect data from patients with and without the disease. The data can include demographic information, medical history, clinical symptoms, and laboratory test results.

2. Clean and preprocess the collected data to remove missing or irrelevant information, normalize the features, and ensure that the data is suitable for the SVM algorithm.

3. Identify the most relevant features that can differentiate between patients with and without the disease using techniques such as correlation analysis or feature ranking.

4. Train the SVM algorithm on the selected features using the labeled data, i.e., data with known outcomes. The SVM algorithm aims to discover the best possible dividing line, known as the decision boundary or hyperplane, that can separate a given dataset into two distinct classes.

5. To evaluate a model's ability to make predictions, performance metrics like accuracy, sensitivity, specificity, and area under the curve can be employed.

6. Validate the trained SVM model on an independent dataset to test its generalizability and applicability to new patient data.

7. Deploy the trained SVM model in clinical practice for disease prediction and monitoring.

B. *Prediction of Diabetes using the above SVM setup*

Clean and preprocess the data, this includes handling missing values, normalizing the data, and transforming categorical variables into numerical ones. Next, the dataset is typically separated into two sets for training and testing. Once the data is prepared, the SVM algorithm is trained on the training dataset. The algorithm uses mathematical optimization to find the best boundary that separates the two classes of data, in this case, patients with diabetes and patients without diabetes. Once the model has been trained, it is tested using the testing dataset to assess its performance. The accuracy, precision, recall, and F1 score of the model are commonly utilized as metrics to evaluate its performance. If the model's performance is not satisfactory, the algorithm can be fine-tuned by adjusting its parameters, such as the kernel type, regularization parameter, and others. Once the model is trained and fine-tuned to achieve an acceptable level of performance, it can be deployed in a real-world setting to predict diabetes in new patients.

C. *Training and Test Dataset Evaluation*

If you want to evaluate the ability of a support vector machine (SVM) model to accurately predict the incidence of a particular disease, it is crucial to divide the available data into two distinct groups: a training set and a test set. To develop the SVM model, the training set is utilized, whereas the test set is utilized to appraise its effectiveness. Assessment of the model's performance can be accomplished by computing several metrics, including precision, recall, accuracy, and F1-score. It's crucial to verify that both sets accurately represent the entire dataset and that there is no duplication between them. One common approach is to use stratified sampling to ensure that the distribution of the target variable is the same in both sets. Finally, it's important to perform cross-validation on a training set to choose optimal hyperparameters for the SVM model, such as the regularization parameter and kernel function.

The below figure shows the Prediction of Diabetes using a Machine Learning algorithm.

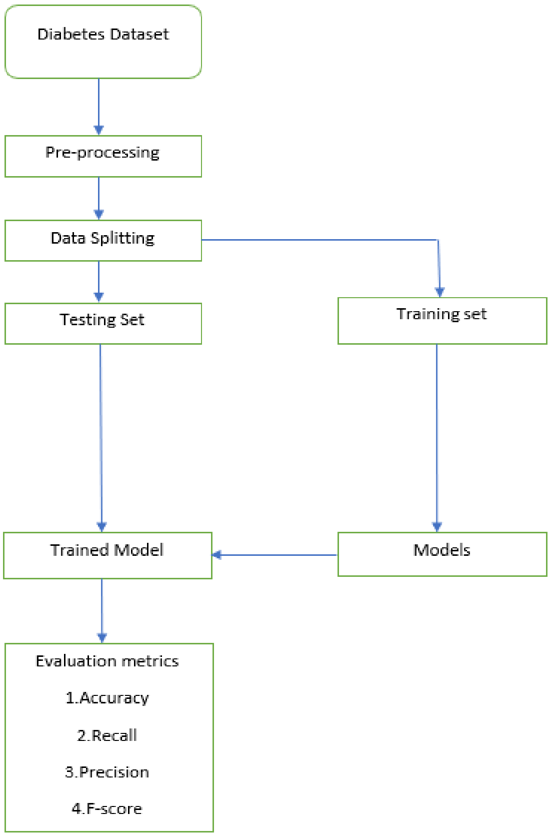


Fig. 1. Model Building using Machine Learning Algorithm

D. *SVM Mathematical Proof:*

1. SVM aims to find a hyperplane that separates the data into different classes while maximizing the margin between the classes.

2. The decision boundary is defined by the equation $w \cdot x + b = 0$, where w is the normal vector, x is the input feature vector, and b is the bias term.

3. The distance between a point x_i and the decision boundary can be calculated as distance $= y\_i \cdot (w \cdot x\_\ddot{\imath} + b)\backslash/ \|w\|$, where y_i is the class label and $|(|w|)|$ is the norm of the weight vector.

4. The goal of SVM is to maximize the margin while ensuring that all training examples lie on the correct side of the decision boundary.

5. This optimization problem can be formulated as minimizing $(1/2)\,|(|w|)|\hat{}2$ subject to the constraints $y\_i \cdot (W \cdot x\_i + b) \geq 1$ for all training examples $(x\_i, y\_i\,)$.

6. The solution to this optimization problem provides the optimal values for the hyperplane parameters w and b, which define the decision boundary.

- *USING DECISION TREE*

A. Experimental Setup of Decision Tree for Diabetes Prediction

1. Collect a dataset of relevant information related to diabetes. This dataset should include various features such as age, BMI, blood pressure, glucose level, etc.

2. Perform data cleaning and preprocessing to remove any missing or erroneous values, normalize the data, and encode categorical features.

3. Split the preprocessed dataset into training and test sets.

4. For selecting the most relevant features, use feature selection techniques that would have the highest impact on predicting diabetes. This could be done using correlation analysis, mutual information, or other statistical methods.

5. Training set is used for building the decision tree model using a suitable algorithm. Experiment with different hyperparameters such as the maximum depth of the tree, minimum samples per leaf, etc. to find the best configuration.

6. Assess the accuracy of the decision tree model when applied to the test data. Utilize measurements like precision, accuracy, recall, and F1 score to evaluate how well the model performed.

7. If the decision tree model's results are not meeting expectations, experiment with different feature selection techniques, hyperparameters, or even try other classification algorithms to improve the performance.

8. Once the model's performance is satisfactory, deploy it in a production environment, such as a web application, where it can be applied to anticipate whether an individual is afflicted with diabetes or not

B. Prediction of Diabetes using Decision Tree setup

Collect the relevant information related to a patient, such as their age, BMI, blood pressure, glucose level, etc. Perform the same data preprocessing steps as used in the experimental setup, such as cleaning and normalizing the data and encoding categorical features. Use the same feature selection techniques as used in the experimental setup to select the most relevant features. Use the prepared patient data as input for the decision tree model to predict whether the patient has diabetes or not. Interpret the decision tree for understanding which features had the most impact on the prediction. This can help in understanding the risk factors for diabetes and identifying potential interventions. If the prediction accuracy is not satisfactory, consider retraining the model using a larger or more representative dataset, experimenting with different feature selection techniques or hyperparameters, or trying a different classification algorithm.

C. Training and Test Dataset Evaluation

The dataset used for training has a collection of labeled instances, where each instance is composed of a group of characteristics or properties and a label that can be one of two possible values which indicate whether the patient has diabetes or not. The decision tree algorithm uses these examples to learn a set of rules that can accurately predict the label of new, unseen examples. Before training the decision tree model, the raw data must be preprocessed. This involves steps such as cleaning and normalizing the data, handling missing values, and encoding categorical variables. Preprocessing is necessary to ensure the decision tree model can accurately learn patterns in the data. The decision tree algorithm uses the preprocessed training dataset and the selected features to build a decision tree model. The algorithm recursively partitions the feature space into smaller and smaller regions based on the training examples, creating a tree structure where each node represents a split on a feature. The decision tree model's effectiveness is assessed by testing it with a dataset that was not used during the training process. This is done to ensure that the model can generalize well to new data. The decision tree model is used to make predictions on the preprocessed test dataset, and then the predicted labels are compared to the true labels to calculate evaluation metrics like precision, accuracy, F1 score, and recall. If the performance given by the decision tree model is not satisfactory, the feature selection process can be revisited, and hyperparameters of the decision tree algorithm can be tuned to improve the model's performance.

D. Decision Tree Mathematical Proof:

1. The decision tree algorithm aims to create a tree structure that makes decisions based on feature values to predict the target variable.

2. At each node of the tree, a decision is made based on a specific feature and its corresponding threshold value.

3. The decision tree algorithm optimizes the tree structure by minimizing a cost function, such as Gini impurity or entropy.

4. Gini impurity (denoted as Gini) measures the impurity or uncertainty of a node and is calculated as follows:

Gini(node) $= 1 - \Sigma(P\_i\,)\hat{}2$

where P_i is the proportion of samples belonging to class i in the node.

5. Entropy (denoted as H) is another measure of impurity and is calculated as:

where P_i is the proportion of samples belonging to class i in the node (excluding 0 values).

6. Information Gain (denoted as IG) “Eq. (1)” is the measure of the decrease in impurity achieved by a split and is calculated as the difference between the impurity of the parent node and the weighted impurity of the child nodes:

$$IG(parent, feature) = impurity(parent) - \Sigma\left(\left(\frac{n_i}{n}\right) \cdot impurity(child_i)\right) \tag{1}$$

where n_i is the number of samples in child node i, n is the total number of samples in the parent node, and $impurity(\dashv)$ represents either Gini impurity or entropy.

7. The decision tree algorithm selects the feature and threshold value that maximize the information gain at each node.

8. The algorithm recursively continues splitting the data based on the selected features until reaching a stopping criterion, such as reaching a maximum tree depth or having a minimum number of samples in each leaf node.

IV. Result and Analysis

Classification experiments were carried out using the Diabetes dataset [15], which consisted of 768 data points. The researchers used 614 of these data points for training the model, while the remaining 154 data points were reserved for testing purposes.

Table 1 shows classification Accuracy Using a Support Vector Machine

Dataset	Samples	Training Data	Attributes	Using SVM
Diabetes	760	614	154	76.6%

Table 1. Classification Accuracy Using SVM

Table 2 shows classification Accuracy Using a Decision Tree

Dataset	Samples	Training Data	Attributes	Using Decision Tree
Diabetes	760	614	154	75%

Table 2. Classification Accuracy Using Decision Tree

a. Confusion Matrix:

Followed fig. 2 and Fig. 3 show Confusion Matrix for SVM and Decision Tree respectively.

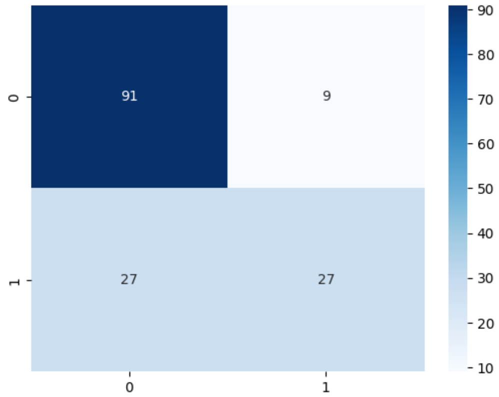


Fig. 2. Confusion matrix for SVM

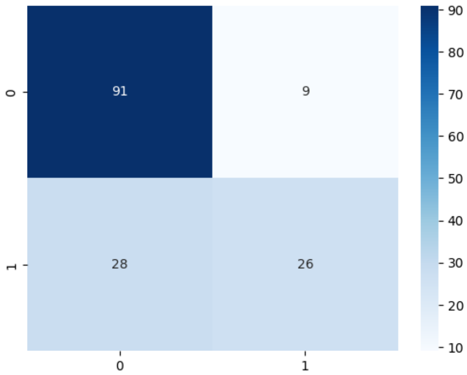


Fig. 3. Confusion Matrix for Decision Tree

b. Performance metrics:

The performance of classification algorithms can be compared across different metrics as shown in Table 3 and its visualization is shown in fig. 4, 5, 6, and 7.

- *Accuracy*: It is a metric used to measure the overall accuracy of a classifier. It shows how accurately the classifier has predicted the result. The accuracy of a classifier is given by the below equation (2)

$$accuracy(a) = \frac{tp + tn}{tp + tn + fp + fn} \qquad (2)$$

- *Precision*: It is a metric used to measure the accuracy of a model's positive predictions. In simpler terms, precision refers to the accuracy of the positive predictions made by the model, which means the proportion of correct positive predictions out of all the positive predictions made. The formula for precision is given by the below equation (3)

$$preciion(a) = \frac{tp}{tp + fp} \qquad (3)$$

- *Recall*: It is a metric used to assess a model's capacity of recognizing all positive examples. More specifically, it represents the proportion of accurate positive predictions made by the model out of all real positive instances present in the data. The formula for the recall is given by the below equation (4)

$$recall(r) = \frac{tp}{tp + fn} \qquad (4)$$

- *F1-score*: It is a metric used to measure a model's accuracy that combines both precision and recall into a single score. This score is derived by computing a weighted average of precision and recall, with a maximum score of 1 and a minimum score of 0. The F1 score is used to assess the model's performance and determine the optimal balance between precision and recall. It is given by the below equation (5)

$$F1\ Score = \frac{tp}{tp + 1/2(fp + fn)} \qquad (5)$$

	Accuracy	Precision	Recall	F1-Score
SVM	0.766	0.75	0.5	0.6
Decision Tree	0.75	0.7428	0.4814	0.5842

Table 3. Performance of Classifier

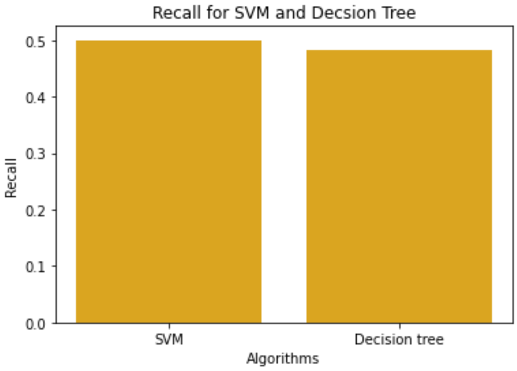


Fig. 4. Recall

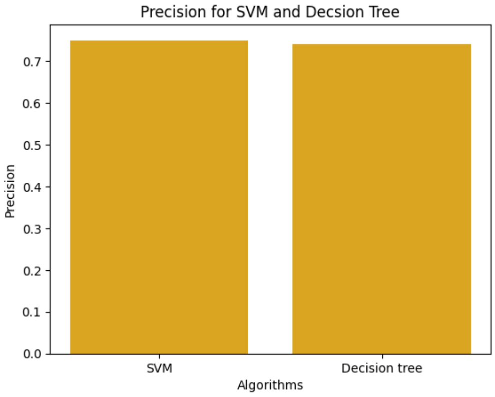


Fig. 5. Precision

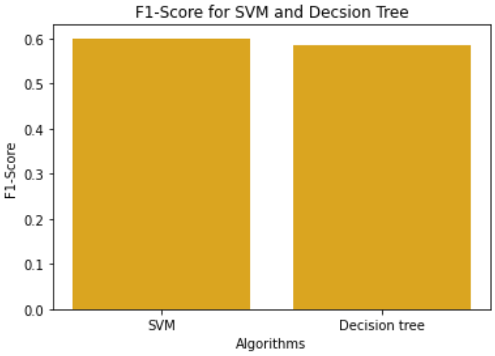


Fig. 6. Score

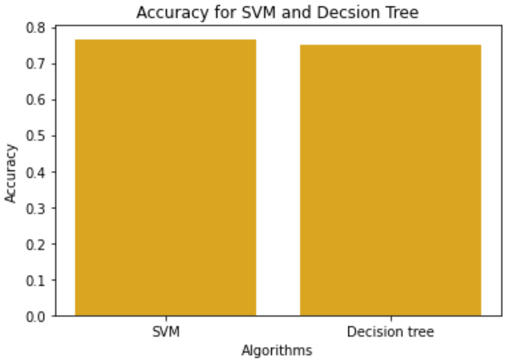
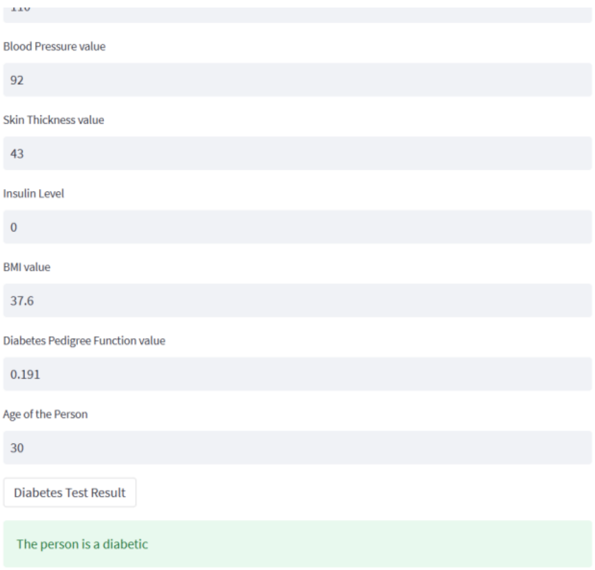


Fig. 7. Accuracy

c. Creating a User Interface for Accessibility

This project aims to forecast the occurrence of diabetes in humans based on the pertinent medical data gathered. The process entails an individual inputting all the necessary medical information into an online portal, and the data is then fed into a pre-trained model that predicts whether the individual has diabetes or not. The accuracy rate of the model's prediction is reported to be 76% as shown in Table 1, which is considered satisfactory and dependable. To input the required medical data, a simple user interface form is provided, which prompts the user to fill in specific medical data fields. The User interface of the website is shown below in fig.8 and 9.



110

Blood Pressure value

92

Skin Thickness value

43

Insulin Level

0

BMI value

37.6

Diabetes Pedigree Function value

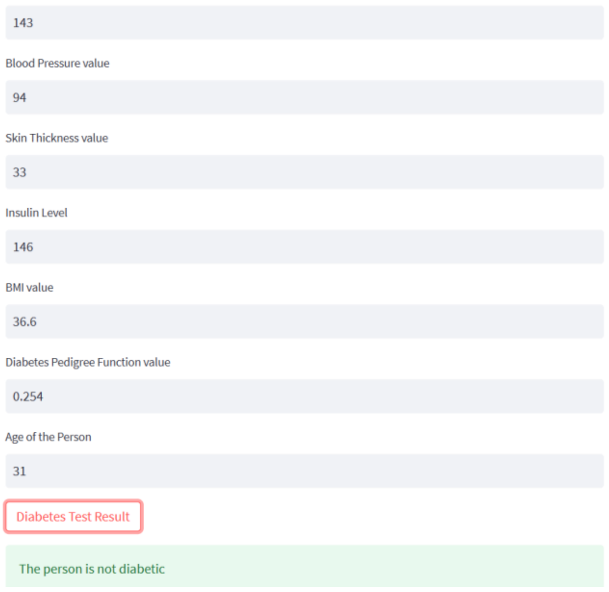
0.191

Age of the Person

30

The person is a diabetic

Fig. 8. Prediction for diabetic person



143

Blood Pressure value

94

Skin Thickness value

33

Insulin Level

146

BMI value

36.6

Diabetes Pedigree Function value

0.254

Age of the Person

31

The person is not diabetic

Fig. 9. Prediction for a non-diabetic person

V. CONCLUSION

This project aims to develop a model that could accurately identify diabetes patients who are at a high risk of being admitted to the hospital. The model analyzed various factors such as blood glucose and insulin levels, age, blood pressure, skin thickness, and BMI by using support vector models and medical record analysis. SVM demonstrates potential as a valuable tool for early detection and management of diabetes, leading to improved patient outcomes. The results showed that the SVM classifier provided the most accurate prediction for the dataset of diabetes patients.

For future work, more complex ML algorithms should be created to improve disease prediction efficiency. The learning models should be calibrated more frequently after the training phase for better performance. Additionally, the datasets should be expanded to include different demographics to avoid overfitting and increase accuracy. Relevant feature selection methods should also be used to enhance the performance of learning models. Once the disease is predicted, the required medical resources could be managed efficiently, resulting in lower costs for treating the disease.

VI. REFERENCES

[1] Kumari, V. Anuja, and R. Chitra. "Classification of diabetes disease using support vector machine." International Journal of Engineering Research and Applications 3.2 (2013): 1797-1801.

[2] Ayon, Safial Islam, and Md Milon Islam. "Diabetes prediction: a deep learning approach." International Journal of Information Engineering and Electronic Business 12.2 (2019): 21.

[3] Hasan, Md Kamrul, et al. "Diabetes prediction using ensembling of different machine learning classifiers." IEEE Access 8 (2020): 76516-76531.

[4] CORTES, C., & VLADIMIR VAPNIK. ((1995)). SupportVector Networks.

[5] Larabi-Marie-Sainte, Souad, et al. "Current techniques for diabetes prediction: review and case study." Applied Sciences 9.21 (2019): 4604.

[6] Phalak, P., K. Bhandari, and R. Sharma. "Analysis of decision tree-a survey." International Journal of Engineering Research 3.3 (2014).

[7] Cheng-Hong, Yang, et al. "Prediction of mortality in the hemodialysis patient with diabetes using support vector machine." Revista Argentina de Clínica Psicológica 29.4 (2020): 219.

[8] Awasthi1, A., Gangwal, I., & Jain, M. (June 2022-). Diabetes Prediction Using Machine Learning

[9] Fregoso-Aparicio, Luis, et al. "Machine learning and deep learning predictive models for type 2 diabetes: a systematic review." Diabetology & Metabolic Syndrome 13.1 (2021): 1-22.

[10] Sharma, Toshita, and Manan Shah. "A comprehensive review of machine learning techniques on diabetes detection." Visual Computing for Industry, Biomedicine, and Art 4 (2021): 1-16.

[11] Breiman, L., et al. "Classification and regression trees (Wadsworth, Belmont, ca, 1984)." Proceedings of the Thirteenth International Conference, Bari, Italy. 1996.

[12 Quinlan, J. Ross. "Combining instance-based and model-based learning." Proceedings of the tenth international conference on machine learning. 1993.

[13] Wu, X, Sun.j, Zhang, Y., & Wang, Y. ((2021)). "An improved support vector machine classifier for breast cancer detection". Computers in Biology and Medicine.

[14] Li, J., Chen, Y.,, & Wang, B. (2016). "An efficient multiclass SVM algorithm for text categorization". International Journal of Computational Intelligence Systems

[15] https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database