Gazi University
Journal of Science
PART A: ENGINEERING AND INNOVATION
http://dergipark.org.tr/gujsa

Comparative Performance Analysis of Selected Machine Learning Algorithms and the Stacking Ensemble Method for Prediction of the Type II Diabetes Disease

Nathan Ayuba ZOAKAH[1*] Augustine Shey NSANG[1] Abel Adeyemi AJIBESIN[1] Ayuba Ibrahim ZOAKAH[2]

[1] School of Information Technology and Computing, American University of Nigeria, Yola
[2] Department of Community Medicine, University of Jos, Nigeria

Keywords

Machine Learning

Diabetes

Support Vector Machine

Stacked Ensemble Method

Abstract

Diabetes is a prevalent non-communicable disease affecting many people globally. The common risk factors are obesity, age, lack of exercise, lifestyle, genetic factors, high blood pressure, and poor diet. Early identification of this condition can help prevent subsequent complications, including heart attacks, lower limb amputations, nerve damage, and blindness. Data mining and machine learning have become popular and successful methods of identifying numerous diseases, including Diabetes, using clinical data over the years. This study focuses on the principles and processes of Naïve Bayes, Support Vector Machines, Logistic Regression, Decision Tree, and Random Forest algorithms for diabetes prediction, using the Scikit-learn inbuilt libraries for the experiments. Furthermore, we ensemble all five machine learning models to produce a single stacked ensemble model. Data preprocessing techniques such as scaling, missing data removal, dimensionality reduction, and balancing of target class were performed on the **Jos Urban Diabetes** dataset used for this study. The comparison of the algorithms' performances across various evaluation metrics, demonstrates that the Support Vector Machines algorithm outperform all others in terms of **Accuracy**, **Precision**, **Sensitivity**, and **Matthew's Correlation Coefficient** with scores of **96.11%**, **91.61%**, **85.67%**, and **82.59%** respectively with 10-fold cross-validation. Furthermore, the Stacked Ensemble Method model had the best **Area Under the Receiver Operating Characteristic Curve** scores of **98.47%** with 10-fold cross-validation.

Cite

Zoakah, N. A., Nsang, A. S., Ajibesin, A. A., & Zoakah, A. I. (2024). Comparative Performance Analysis of Selected Machine Learning Algorithms and the Stacking Ensemble Method for Prediction of the Type II Diabetes Disease. *GU J Sci, Part A, 11*(3), 622-646. doi:10.54287/gujsa.1531997

Author ID (ORCID Number)		Article Process	
0009-0000-3873-8471	Nathan Ayuba ZOAKAH	**Submission Date**	12.08.2024
0000-0002-6466-9032	Augustine Shey NSANG	**Revision Date**	09.09.2024
0000-0001-6518-0231	Abel Adeyemi AJIBESIN	**Accepted Date**	16.09.2024
0000-0003-1856-7753	Ayuba Ibrahim ZOAKAH	**Published Date**	30.09.2024

1. INTRODUCTION

Diabetes mellitus is a chronic disorder that arises when the body does not produce enough insulin or utilize the insulin produced adequately (IDF, 2021). It is a disorder that develops when the blood sugar or glucose in the blood reaches abnormal high levels (hyperglycemia) due to its inability to reach other cells in the body (Birjais et al., 2019). Diabetes is of two main types: Type I and Type II. Other kinds of Diabetes include monogenic Diabetes (formerly known as secondary Diabetes), gestational Diabetes, and other unclassified Diabetes.

Type I Diabetes occurs when the endocrine gland in the pancreas produces little or no insulin required to regulate the blood sugar level. It is caused by an autoimmune reaction in which the body's immune system affects the pancreatic insulin-producing beta-cells. This type of Diabetes is most common in children and adolescents (Choudhury & Gupta, 2019). Although the initial causes of this type of Diabetes are unknown, research shows that the cause of this autoimmune reaction may be the combination of environmental triggers like viral infections and genetic susceptibility (IDF, 2021).

*Corresponding Author, e-mail: nathanzoakah@gmail.com

Persons with type I Diabetes will require a daily dosage of insulin injections, regular glucose level monitoring, and support to live healthy lives and avert further complications. Some symptoms of type I Diabetes are constant hunger, feelings of fatigue, blurred vision, and diabetic ketoacidosis (DKA).

Type II diabetes, also known as "insulin resistance," occurs when the endocrine gland in the pancreas produces insulin that the body cannot adequately use to regulate the blood sugar level. Ninety percent (90%) of Diabetes worldwide is said to be type II diabetes making it the most common type of Diabetes (IDF, 2021). The reasons for type II diabetes are unknown; however, there is a clear association between being overweight or obese, becoming older, ethnicity, and having a family history. Although they have similar symptoms with type I diabetes, they tend to be less dramatic and sometimes show no symptoms. However, some benchmarks would enable someone with type II diabetes to watch for leading long, healthy lives and avert long-term complications. These include a healthy diet, physical activities, maintaining body weight, smoking cessation, regular checkups, and taking prescribed medication by medical personnel.

Blood glucose levels higher than usual but not yet high enough to be identified as type II diabetes are **prediabetes** (Punthakee et al., 2018).

Diabetes is a major public health problem. According to World Health Organization (WHO, 2021), around 1.5 million people worldwide died directly from Diabetes in 2019, making it the ninth-highest cause of death. Furthermore, according to the International Diabetes Federation (IDF), Diabetes affected 415 million people in 2015, which is anticipated to climb to 642 million by 2040 (Zimmet et al., 2016). In a recent report by the IDF (2021), Diabetes is shown to be one of the fastest-growing emergencies of the 21st century, as revealed in the following statistics. About 537 million people had Diabetes in 2021; the projected numbers for 2030 and 2045 are 643 million and 783 million, respectively. In 2021 alone, more than 1.2 million children and adolescents had type I diabetes.

Diabetes can cause a variety of dangerous long-term complications, including cardiovascular diseases, stroke, renal failure or eye damage (retinopathy), heart attack, lower limb amputation, kidney damage (nephropathy), peripheral artery disease, blood vessels, and nerve damage (neuropathy) (Tigga & Garg, 2020). However, Diabetes and all its associated problems can be significantly reduced or prevented if it is detected, treated early, and appropriately managed.

Machine learning and Data Mining techniques have been used in the medical domain as very reliable tools for predicting Diabetes from clinical data. Data mining is extracting information from data and uncovering numerous patterns inherent in the data that are accurate, novel, and beneficial (Bhatia, 2019). This process helps to uncover hidden trends in a vast amount of data to support decision-making. On the other hand, Machine Learning is a branch of Artificial Intelligence that enables computers to learn from data without being specifically programmed to do so (Bhatia, 2019). It uses various algorithms to make predictions from the data prepared through data mining processes with little or no human intervention. Machine learning aims to create a computer program that can access data and utilize it to learn (Ibrahim & Abdulazeez, 2021).

The usage of data mining has accelerated in the Big Data era. With their power and automation, data mining technologies can handle massive volumes of data and extract value (Shmueli et al., 2019). Health practitioners are progressively moving health and healthcare data from traditional to digital formats, and as a result, healthcare institutions are creating vast quantities of data (Armstrong, 2022). However, as with many real-world data, they are prone to inconsistencies and errors such as missing or noisy data. Data preprocessing is a preliminary step in the data mining and machine learning process, which helps to eliminate or reduce such inconsistencies in data. In addition, data quality is a crucial consideration in the data mining process for disease prediction and diagnosis since poor data quality might lead to erroneous or low prediction results during machine learning (Zhu et al., 2019).

With the advent of E-health records and databases and the massive quantity of data collected from hospitals and medical records, early identification of Diabetes is achievable through predictive analysis. This can be done through a physician's knowledge and expertise with the illness; nevertheless, such work is prone to mistakes and inaccuracies if performed manually, depriving the patient of adequate therapy (Khanam & Foo, 2021). Automating this process using machine learning and data mining can reveal hidden patterns in the data, allowing for better decision-making.

Therefore, the purpose of this research is to understand the basic principles of the selected ML algorithms, implement the algorithms on the diabetes dataset and, evaluating the algorithms performances while following best practices and important data mining and machine learning steps to achieve the expected results.

2. MATERIAL AND METHOD

In this research, we employed an ensemble approach and supervised machine learning techniques to evaluate the performance of five algorithms. In supervised learning, algorithms are trained on labeled data to generate predictions based on input data. This method learns a mapping between the input and output data, using pairs from a labeled dataset to understand this relationship. To accurately predict new, unknown data, the algorithm attempts to comprehend the link between the input and output data. Techniques such as regression and classification are incorporated.

For this study, binary classification was utilized to categorize the diabetes dataset into diabetic and non-diabetic (control) groups. The methodology comprised six major steps:

1. Data Collection: Gathering relevant data for analysis.
2. Exploratory Data Analysis: Examining the data to uncover patterns and insights.
3. Data Processing: Preprocessing the data, including handling missing values, scaling, and dimensionality reduction.
4. Machine Learning: Training the algorithms on the processed data.
5. Ensemble Learning: Combining multiple models to improve performance.
6. Performance Evaluation: Assessing the accuracy and effectiveness of the models.

Each of these steps is crucial for ensuring the robustness and reliability of the machine learning models.

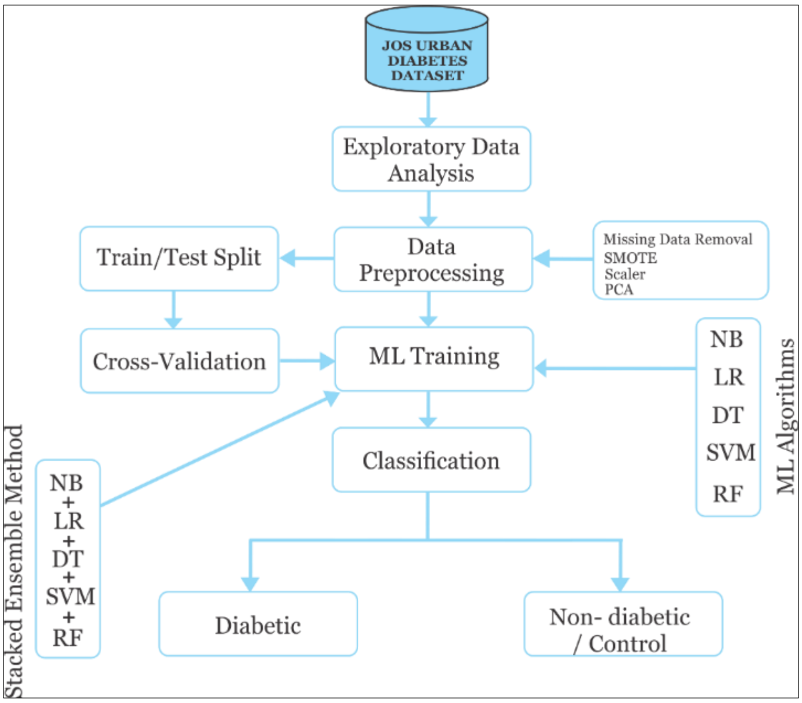


Figure 1. *Overview of the General Methodological Workflow*

Data Collection Phase

The Apex Clinic and Diabetes Screening Centre in Jos, Plateau State, Nigeria, owns the dataset utilized in this study. This data was gathered and recorded during routine screenings of individuals visiting the diabetes clinic mentioned above. Age, Fasting Plasma Glucose, Diastolic blood pressure, Systolic blood pressure, Weight, Height, Body Mass Index, Waist Circumference, Hip Circumference, and Gender were all recorded. The features of the diagnostic indicators or variables in the Jos Urban Diabetes dataset are shown in Table 1. This dataset comprises 753 records from various people gathered from the aforementioned diabetic clinic. For each attribute: (numeric-valued and strings) containing 753 observations.

Table 1. *Description of the Jos Urban Diabetes Dataset*

S/N	Attribute Name / Measurement	Attribute Data type	Range
1	Age (years)	Integer	10 – 85
2	Fasting Plasma Glucose (mg/ dl)	Integer	59 – 495
3	Diastolic blood pressure (mm Hg)	Integer	89 – 217
4	Systolic blood pressure (mm Hg)	Integer	26 – 148
5	Weight (Kilogram)	Integer	28 – 124
6	Height (Meters)	Float	1.30 – 1.96
7	Height Squared	Float	1.69 – 3.84
8	Body mass index (weight in kg / (height in m)^2)	Float	14.90 – 49.90
9	Waist Circumference (Centimeters)	Float	21.00 – 156.00
10	Hip Circumference (Centimeters)	Float	61.00 – 143.00
11	Waist Hip Ratio	Object or String	0.24 – 1.24
12	Gender	Object or String	Male and Female
13	Diagnosis Class Distribution: (class value 1 is interpreted as "tested positive for diabetes" and class value 0 is interpreted as “tested negative”)	Integer	Class variable (0 or 1)

Exploratory Data Analysis Phase

The primary goal of this phase is to comprehend the dataset utilized for this study. Various graphical and non-graphical visualization techniques were used to identify missing values, features, the correlation between features, target variables, the data type of the variables, the shape of the dataset, and other statistical attributes such as mean and standard deviation.

Data Preprocessing Phase

In this phase, various techniques were used to handle inconsistencies in the dataset identified in the previous phase. The data was scaled and transformed; missing data values were removed and the imbalanced target variable was balanced using a resampling technique. The dataset was split into two (training and testing sets). After that, the data was ready to be sent to the next phase.

Machine Learning Phase

During this phase, five machine learning algorithms were implemented. The training dataset used Naïve Bayes, Logistic Regression, Support Vector Machine, Decision Tree, and Random Forest algorithms to produce a prediction model or classifier. All of the algorithms stated above used the inbuilt sci-kit learn ML algorithms in this study.

Ensemble Learning Phase

All five algorithms used in this article were combined in this phase to generate a single stacked model. To learn from each of the models built by the different algorithms employed, a final estimator or classifier is used. The final estimator utilized for the ensemble approach to build the stacked model was the Logistic Regression algorithm from the sci-kit learn module.

Performance Evaluation Phase

The testing dataset was used to evaluate the performance of the different machine learning algorithms or classifiers in this phase of the proposed methodology. The accuracy, precision, sensitivity, MCC, and ROC area under the curve were then used to compare the findings.

Figure 2 provides a detailed description of the methodological workflow, beginning with data collection and exploratory data analysis. The first step in data preprocessing was the removal of missing data, followed by splitting the dataset into training and testing sets. Additional preprocessing steps included class balancing using SMOTE, scaling, and dimensionality reduction via PCA for the feature variables and target variable of the training dataset. For the testing dataset, only scaling and dimensionality reduction were performed.

Furthermore, 10-fold cross-validation was conducted on both splits of the dataset. The training dataset was utilized for the learning process with all five machine learning algorithms, including hyperparameter tuning and the stacked ensemble method. The trained classifiers or models were then applied to the testing dataset to make classifications and measure the performance of each algorithm. Finally, the best classifiers or models were saved.

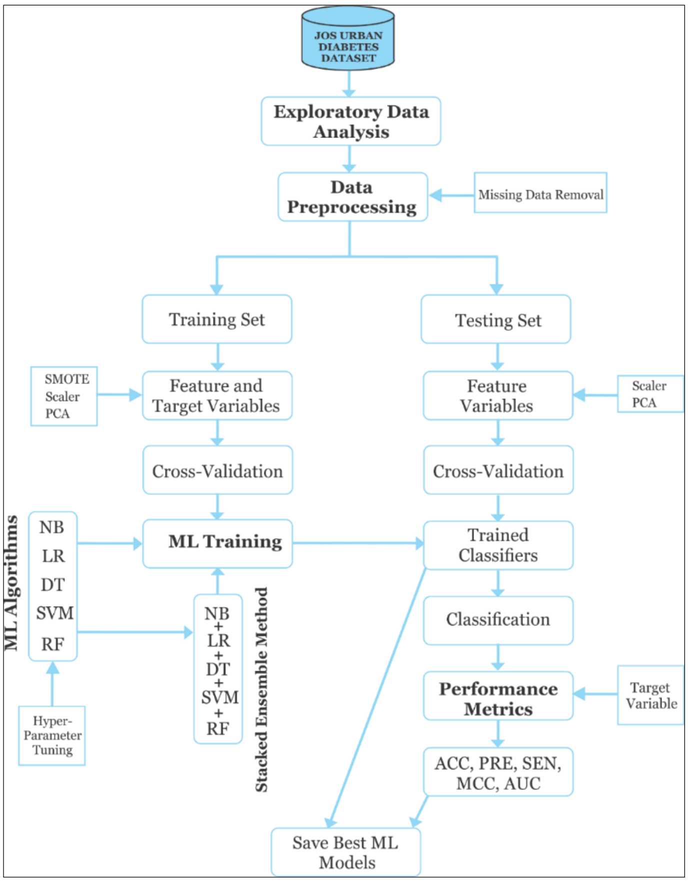


***Figure 2**. Detailed Methodological Workflow Diagram*

SELECTED MACHINE LEARNING ALGORITHMS

The five selected supervised learning algorithms are discussed in this section. Some of the mathematical and statistical foundations as well as how these algorithms learn from data are also briefly explained.

Naïve Bayes Algorithm

The Naïve Bayes algorithm is a classification method that employs Bayes' theorem under the strong assumption that all variables or predictors are independent of one another (Prasanna, 2019). It indicates that features inside a class are unrelated to other features within the same class. Another strong assumption with this algorithm is that all the variables or predictors have an equal effect on the target variable (Gandhi, 2018). It can be used for either binary or multiclass classification problems. When using a Bayesian classification, our main aim is to find the probability of a label (L) given some observed features (F), also known as the posterior probability $\boldsymbol{P(L \mid F)}$. It can be expressed using Bayes' Theorem as shown in the formula below:

$$\boldsymbol{P\,(L\,|F)} = \frac{\boldsymbol{P\,(L)\,P\,(F \mid L)}}{\boldsymbol{P(F)}} \qquad (\mathbf{1})$$

Where;

$\boldsymbol{P(L\,|F)}$ is the posterior probability

$\boldsymbol{P(L)}$ is the class prior probability

$\boldsymbol{P(F|L)}$ is the class conditional probability or likelihood of a predictor given a class

$\boldsymbol{P(F)}$ is the predictor of prior probability or evidence

The feature vector **F** can be written as

$$\boldsymbol{F = (f_1, f_2, f_3, \dots, f_n)}$$

where $\boldsymbol{f_1, f_2, f_3, \dots, f_n}$ correspond to the feature variables in a dataset.

Based on the assumption that all the features are mutually independent, we substitute **F** in equation **(1)** and expand using Chain's rule. The resulting equation is shown in **(2)**:

$$\boldsymbol{P\,(L \mid f_1, \dots, f_n)} = \frac{\boldsymbol{P\,(f_1 \mid L) * \; P\,(f_2 \mid L) * \; \dots * P(f_n|L)\,.\; P\,(L)}}{\boldsymbol{P(f_1) \; * \; P(f_2) * \dots * \; P(f_n)}} \qquad (\mathbf{2})$$

Equation **(3)** is used to find the class with the highest probability and demonstrates how the label (L) value is calculated given the features.

$$\mathbf{L} = \mathbf{argmax}_{\boldsymbol{L}}\, \boldsymbol{P\,(f_1 \mid L) * \; P\,(f_2 \mid L) * \; \dots * P(f_n|L)\,.\; P\,(L)} \qquad (\mathbf{3})$$

Because the class probability for each feature is between the range 0 to 1, the values, when multiplied together, give minimal values, which can lead to overflow problems. The solution is to apply a Log function to each class probability ($P\,(f_i \mid L)$), and according to the logarithm law, the multiplication signs change to addition, as seen in equation **(4)** .

$$\mathbf{L} = \mathbf{argmax}_{\boldsymbol{L}}\, \mathbf{Log}\, \boldsymbol{P\,(f_1 \mid L) + \; Log\, P\,(f_2 \mid L) + \; \dots + P(f_n|L) + Log\, P\,(L)} \qquad (\mathbf{4})$$

The prior probability P(L) is the frequency of the class in the training sample; that is,

$$P(L) \; = \frac{\textbf{Number of samples with class label L}}{\textbf{Total number of samples}}$$

To calculate the class conditional probability, it is modeled with the Gaussian distribution formula, as seen in equation **(5)**.

$$\mathbf{P}\ (\boldsymbol{f_i}\ |\mathbf{L}) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}\boldsymbol{\pi}\boldsymbol{\sigma}_l^2}} * \boldsymbol{e}^{-\left(\frac{(\boldsymbol{f_i} - \boldsymbol{\mu_l})^2}{\mathbf{2}\boldsymbol{\sigma}_l^2}\right)} \tag{5}$$

where:

$\boldsymbol{f_i}$ are the feature variables

μ is the mean value of all feature variables in a given class

σ is the variance of all feature variables in a given class

$\boldsymbol{\pi}$ is a constant value of 3.142

e is a constant value of 2.7183

Finally, in equation **(4)**, the class conditional probability and prior probability results are substituted to obtain the posterior probability for each class. The argmax method finds and returns the class with the highest probability (Loeber, 2019a).

Decision Tree Algorithm

A decision tree is generally a binary tree that recursively splits a dataset until we are left with only pure leaf nodes, that is, data with only one type of class (homogeneous class) (Normalized Nerd, 2021). The decision tree comprises two entities, the decision node (parent) and the leaf node (child). Decision nodes contain a condition to split data into leaf nodes, while leaf nodes are used to decide the class of a new data point. The trees choose the decision node based on a statistical calculation called information gain, where the information gain of a node is measured by the Entropy or Gini Index (Pranto et al., 2020).

To calculate the entropy (**E**), also known as the measure of uncertainty, we use the formula below.

$$\boldsymbol{E} = -\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{n}} \boldsymbol{p}(\boldsymbol{X_i}) * \boldsymbol{Log_2 p}((\boldsymbol{X_i})) \tag{6}$$

where $\boldsymbol{p}(\boldsymbol{X}) = \frac{\boldsymbol{Frequency\ of\ \ the\ class\ labels}}{\boldsymbol{Total\ number\ of\ samples}}$

After computing the entropy of the parent node, the data is split into child nodes, and the entropy of the child nodes is computed to determine how much information was acquired after splitting. This measure is known as the information gain (**IG**) and is calculated using the formula below.

$$\boldsymbol{IG} = \boldsymbol{E}(\boldsymbol{parent}) - [\boldsymbol{weighted\ average}] * \boldsymbol{E}(\boldsymbol{children}) \tag{7}$$

Maniruzzaman et al. (2020) gives a brief description of the steps taken to build a decision tree:

1. Build a tree with its nodes as input features;
2. Choose the feature that delivers the most significant information gain when predicting the output from the input features to forecast the output;

3. Repeat the previous stages to create subtrees based on characteristics not utilized in the previous nodes.

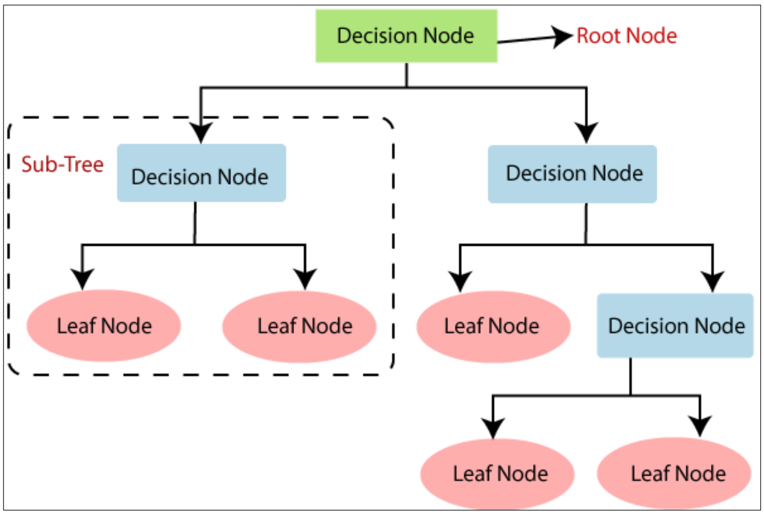


Figure 1. *Graphical Representation of Decision Tree.*

Loeber (2019b) gives a more extended explanation of the procedures necessary for creating the decision tree algorithm. He starts with the steps to construct the tree when training the algorithm and continues with the actions required to predict a new data point when traversing the constructed tree.

Training Phase

1. Begin at the top node and pick the optimal split based on the best information gained at each node.
2. Loop through all features and thresholds (all possible feature values)
3. At each node, save the best-split feature and split the threshold.
4. Recursively generate the tree.
5. To stop the tree from growing, apply certain stopping conditions (max depth, min sample at node, or no more distribution in node (homogeneous class))
6. When we have a leaf node, save its most common label.

Testing Phase (Prediction)

1. Recursively navigate the tree.
2. Look at the best-split feature of the test feature vector F at each node and move left or right based on whether F [feature index] equals the threshold.

When we reach the leaf node, we return the most common class label that was previously saved.

Random Forest Algorithm

Random forest is one of the most well-known and powerful supervised machine learning techniques that is based on ensemble learning, where several classifiers are combined to tackle complicated problems and enhance model accuracy. It is a classifier that uses several decision trees on distinct subsets of a given dataset and averages their results to improve the projected accuracy of that dataset. The new dataset or testing data is disseminated to all newly generated subtrees in the random forest. Each decision subtree in the forest is free to choose the dataset's class (Ibrahim & Abdulazeez, 2021). The model will then identify the best-suited class based on majority voting. Random Forest can be applied in various biomedical studies, particularly in diagnosing diabetes (Maniruzzaman et al., 2020). The following are simplified stages demonstrating how the random forest algorithm works (Sruthi, 2021). Figure 4 shows the graphical illustration of the random forest algorithm.

1. In a Random Forest, n records are chosen randomly from a data collection of k records.
2. An independent decision tree is constructed for each sample (The same procedure is used in the decision tree algorithm above).
3. Each decision tree will provide a result.
4. Majority Voting determines the resolution of classification problems.

Bootstrapping is a random sampling procedure or approach used in the random forest algorithm in which the algorithm is trained on only a portion of the observations rather than all. The subtrees' outputs are aggregated, referred to as bagging when combined with the bootstrapping procedure (Choudhary, 2021).

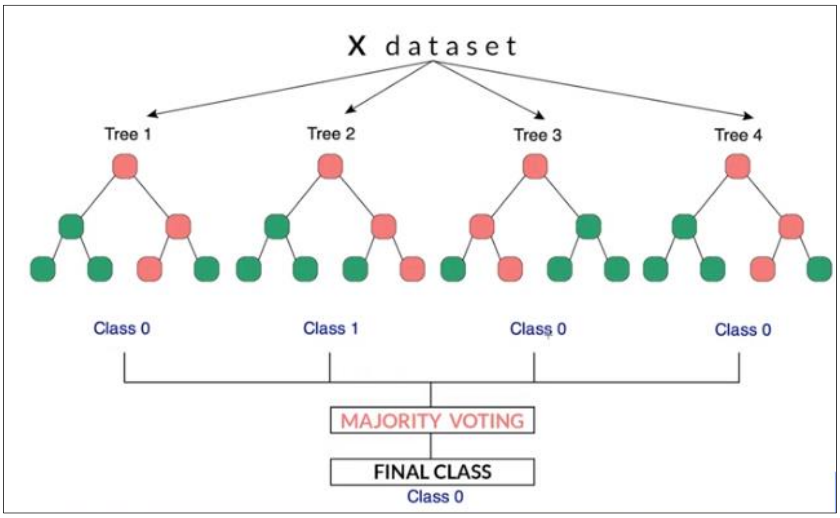


Figure 2. *Graphical Representation of Random Forest Algorithm (Loeber, 2019c)*

Support Vector Machine Algorithm

SVM is a supervised machine modeling technique that is widely used in classification (Sisodia & Sisodia, 2018). It is an algorithm that employs the hypothesis space of linear functions in a high-dimensional feature space and is taught with a learning algorithm from optimization theory that applies a learning bias derived from statistical learning theory (Jakkula, 2006). SVM aims to use an appropriate hyperplane (linear decision boundary) to classify data points in a multidimensional space. A *hyperplane* is a decision boundary used to categorize data points. A hyperplane far away from the data points in each category should be chosen; the farther our data points are from the hyperplane, the more confident we are that they have been correctly recognized. The support vectors are the spots closest to the classifier's margin (Tigga & Garg, 2020). The following explanation is limited to linear separable SVM models with linear kernels. Nonlinear SVM models include kernels such as polynomial and radial basis function kernels.

The linear model is represented in Equation **(8)**

$$\boldsymbol{W * X - b = 0} \tag{8}$$

The function should satisfy the following criteria to determine which class a data point belongs to, as shown in Figure 5 (positive red class or negative blue class) and as illustrated in the following equations **(9)** and **(10)** below.

$$\boldsymbol{W * X_i - b \geq 1} \qquad \boldsymbol{if\ \ y_i = 1} \tag{9}$$

$$\boldsymbol{W * X_i - b \leq -1} \qquad \boldsymbol{if\ \ y_i = -1} \tag{10}$$

where:

W = weight vector

$\mathbf{X_i}$ = feature label

b = bias

$\mathbf{y_i}$ = class labels

So, multiplying our class label with our linear model gives us a single equation representing the condition to be satisfied, as shown in equation **(11)**.

$$\boldsymbol{y_i\,(W * X_i - b) \geq 1} \tag{11}$$

Furthermore, to maximize the margins between the two classes represented in Figure 5, we use the formula $\frac{\mathbf{2}}{||\boldsymbol{W}||}$. It is equivalent to minimizing the distance using the formula $\frac{||\boldsymbol{W}||}{\mathbf{2}}$ (Sisodia & Sisodia, 2018). Where **||W||** is the magnitude of the weight vector.

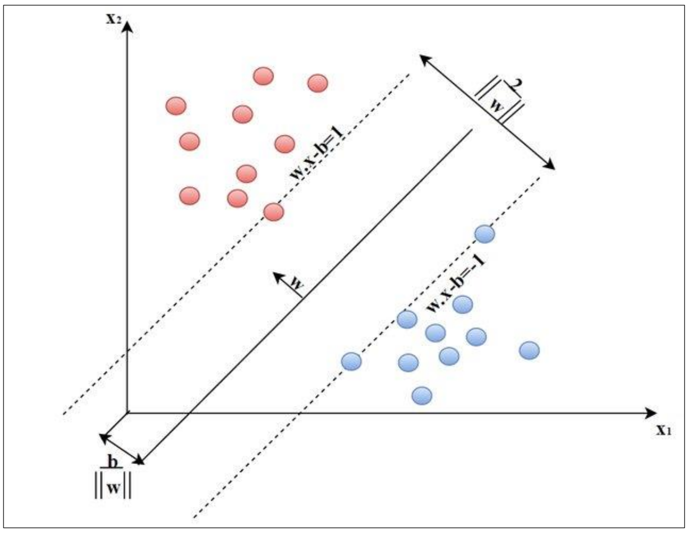


Figure 3. *Support Vector Machine Classifying a Binary Task*

Next, we describe the main objective function, which is made up of two parts: the one responsible for maximizing margins with an extra regularization parameter and the part responsible for establishing the separating hyperplane with a loss function termed Hinge loss (Lanhenke, 2022). Equation **(12)** shows the primary objective function.

$$\boldsymbol{\mathcal{J} = \lambda\,||W||^2 + \frac{1}{n}\sum_{i=1}^{n} \max\big(0, 1 - y_i\,(W * X_i - b)\big)} \tag{12}$$

where:

$\boldsymbol{l = \max\,(0, 1 - y_i\,(W * X_i - b))}$ is the lost function called hinge loss.

and

$\boldsymbol{\lambda\,||W||^2}$ is the magnitude of the **squared Weight vector** multiplied by the regularization parameter **lambda.**

Both parts of the objective function work together to acquire the correct classification of our labels or classes by placing them on the correct side of the hyperplane and having a hyperplane with maximum margins.

The loss function is then optimized using gradient descent by identifying the derivatives for the weights (W) and biases (b). Based on the condition in equation **(11)**, we divide the objective function in equation **(12)** into two classes.

If $\boldsymbol{y_i\,(W * X_i - b) \geq 1}$

Then

$$\boldsymbol{\mathcal{J}_i = \lambda \left|\left|W\right|\right|^2} \tag{13}$$

Else

$$\boldsymbol{\mathcal{J}_i = \lambda\, ||W||^2 + 1 - y_i\,(W * X_i - b)} \tag{14}$$

To obtain the gradients, we apply the condition in equation **(11)** to the derivatives for both classes in equations **(13)** and **(14)**, which is as follows:

If $\boldsymbol{y_i\,(W * X_i - b) \geq 1}$

Then

$$\boldsymbol{\frac{d\mathcal{J}_i}{dW_k} = 2\lambda W_k} \tag{15}$$

$$\boldsymbol{\frac{d\mathcal{J}_i}{db} = 0} \tag{16}$$

Else

$$\boldsymbol{\frac{d\mathcal{J}_i}{dW_k} = 2\lambda W_k - y_i * X_i} \tag{17}$$

$$\boldsymbol{\frac{d\mathcal{J}_i}{db} = y_i} \tag{18}$$

Finally, the weights and biases are updated with the formulas shown in equations **(19)** and **(20)**:

$$\boldsymbol{W = W - \alpha.dw} \tag{19}$$

$$\boldsymbol{b = b - \alpha.db} \tag{20}$$

where:

W = Weight

b = bias

$\boldsymbol{\alpha}$ = learning rate

dw = derivative of the weight and

db = derivate of the bias

Logistic Regression Algorithm

Logistic Regression is another supervised learning algorithm used for solving classification problems. It is a statistical model also known as a logit model which was borrowed into the machine learning field (Birjais et al., 2019). It is a regression model that predicts whether a given data item or entry (feature variables) falls within a certain class (target variable) (Ibrahim & Abdulazeez, 2021). This algorithm uses a Logistic function also known as a Sigmoid function to model a binary classification problem as in our case, predicting if a person is diabetic or not.

In Logistic Regression, we find the probabilistic values that lie between 0 and 1 by applying the sigmoid function in equation **(22)** to our linear model in equation **(21)** which models the probability of our data. Figure 6 shows the sigmoid function graph and Figure 7 shows the probabilistic values, threshold, and maximum values of 0 and 1.

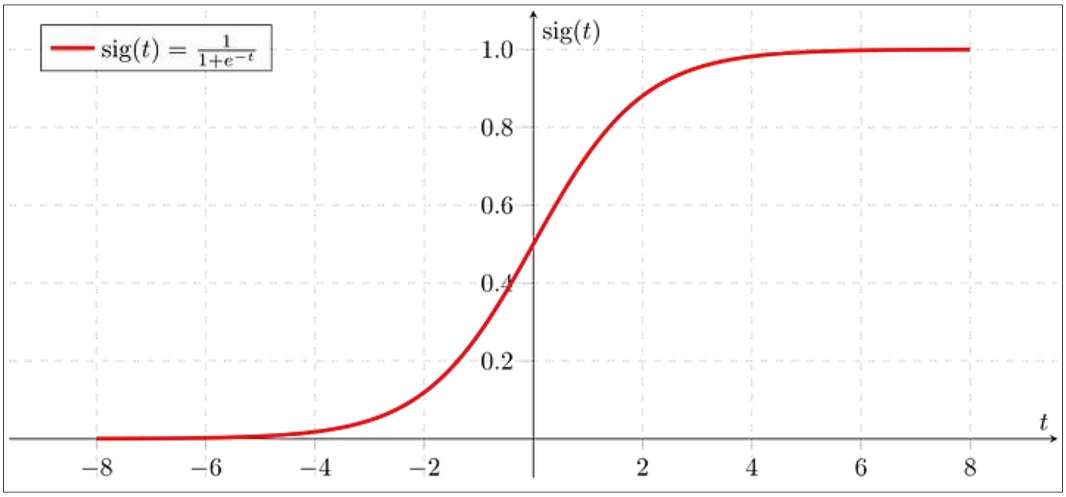


Figure 4. *Sigmoid Function Graph*

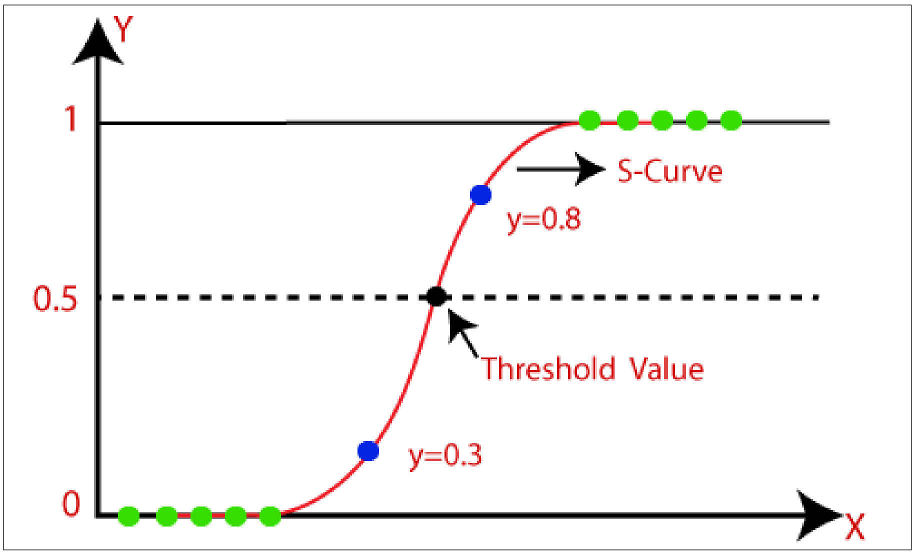


Figure 5. *Probabilistic Values, Threshold, and Maximum Values on the Sigmoid Graph*

The equations for the linear model and sigmoid function are shown below.

$$\boldsymbol{Z = W * X + b} \qquad (21)$$

$$\boldsymbol{S(x) = \frac{1}{1 + e^{-(x)}}} \qquad (22)$$

The sigmoid function is applied to the linear model to determine the approximation of our target variable y, producing equation **(23)**.

$$\hat{y} \quad = S(Z) = \frac{1}{1 + e^{-(W*X+b)}} \qquad (23)$$

Gradient descent, similar to the Support Vector Machine Algorithm described above, is used to optimize the cost function in terms of weight (W) and bias (b). Cross Entropy is the cost function used in logistic regression, as indicated in equation **(24)** (Swaminathan, 2019).

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{n} y_i \log\left(h_\theta(x_i)\right) + (1 - y_i)\log(1 - h_\theta(x_i)\,) \qquad (24)$$

If $y_i = 1$, then the $(1 - y_i)$ term will become zero hence,$\log\left(h_\theta(x_i)\right)$alone will be left.

If $y_i = 0$, then the (y_i) term will become zero hence, log $(1 - h_\theta(x_i)$ alone will be left.

Gradient descent begins at a certain position and iteratively updates the parameters (that is, weight and bias) by computing derivatives and advancing in the direction of the gradient (negative direction) depending on a specific learning rate ($\boldsymbol{\alpha}$).

The derivatives for the weight and bias are as follows;

$$\frac{dJ}{dw} = \frac{1}{N}\sum x_i\,(\hat{y} - \ y_i) \qquad (25)$$

$$\frac{dJ}{db} = \frac{1}{N}\sum (\hat{y} - \ y_i) \qquad (26)$$

Finally, the weights and biases are updated using the formulae shown in equations **(27)** and **(28)** :

$$W = W - \ \alpha.dw \qquad (27)$$

$$b = b - \ \alpha.db \qquad (28)$$

where:

W = Weight

b = bias

$\boldsymbol{\alpha}$ = learning rate

dw = derivative of the weight and

db = derivative of the bias

Stacking Ensemble Method

This is a Stacked Generalization ensemble learning approach (Zhang & Ma, 2012). It is a strategy whose primary goal is to get the best generalization accuracy possible (Rokach, 2009). This is accomplished by combining predictions from many models to form a new model used to generate predictions on the test dataset (Singh, 2021). The diagram below depicts the general graphical illustration of stacking.

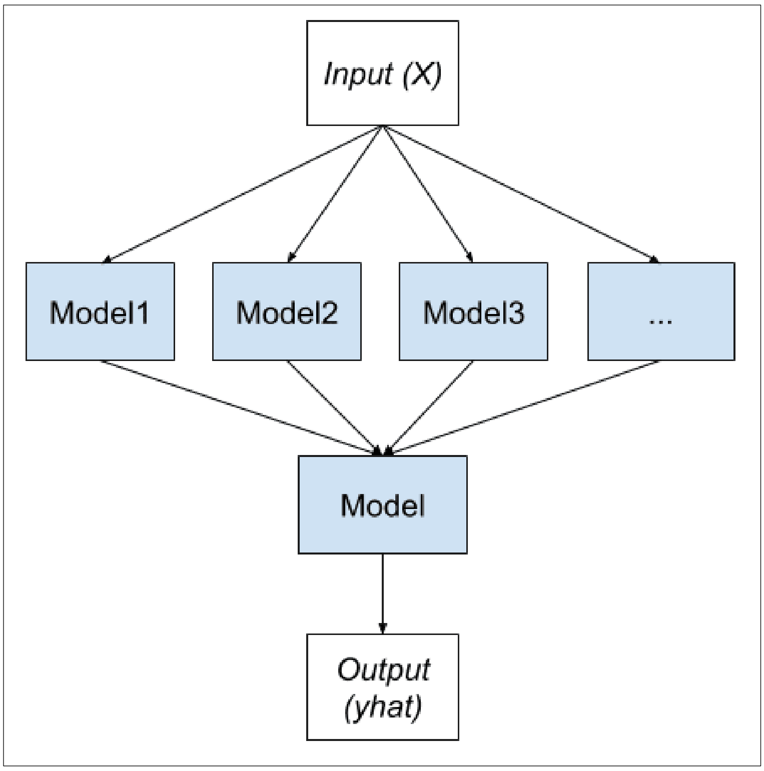


Figure 6. *General Stacking Illustration*

Harrison (2022) provided a more detailed understanding of the stacking ensemble learning approach in the stages below.

1. The dataset is divided into two parts: training and testing.
2. The stacking technique starts with individual model training, sometimes known as the level 0 model stage. Each model predicts on the training dataset, and the outcomes of each model prediction (target class) are added to the training dataset. This modifies the training dataset by adding new feature variables.

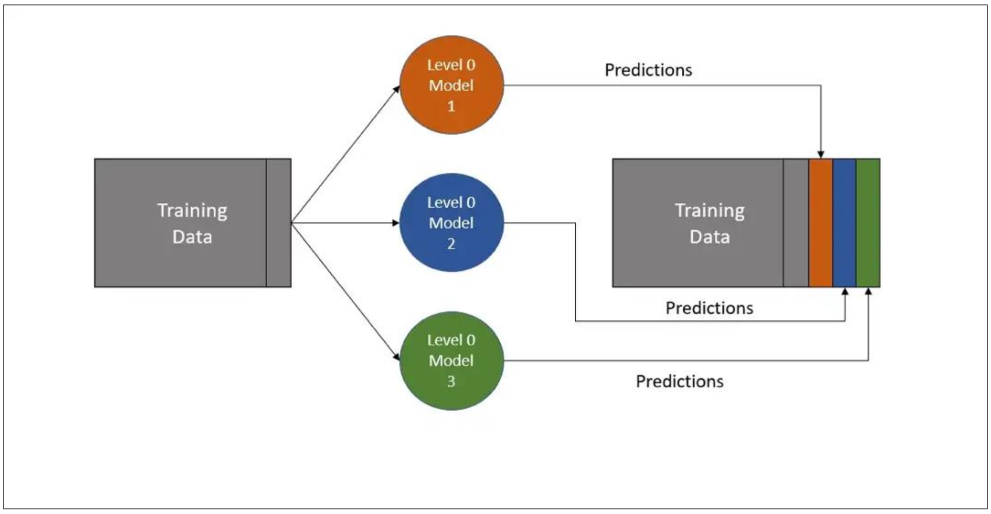


Figure 7. *Level 0 Stacking Stage (Harrison, 2022).*

3. The newly altered training dataset is then trained on another model, referred to as the level 1 model stage. This stage generates the trained stacking model, often known as the level 1 model.

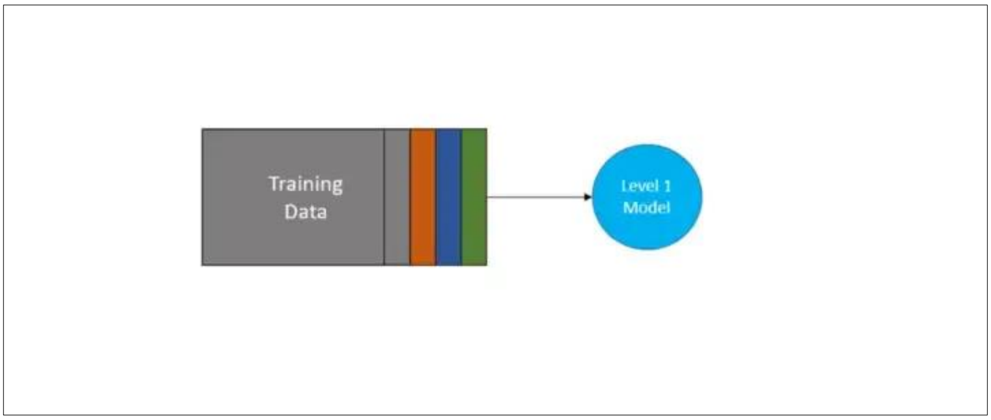


Figure 8. *Level 1 Stacking Stage (Harrison, 2022).*

4. The last step makes final predictions using the level 1 model on the testing dataset. But first, the testing dataset must be changed to have the same shape as the modified training dataset. As a result, each level 0 trained model is applied to the testing dataset to generate new predictions, which are then added to the testing dataset, resulting in the modified testing dataset. Finally, the stacked model is utilized to produce the final prediction using the trained level 1 model on the modified testing dataset.

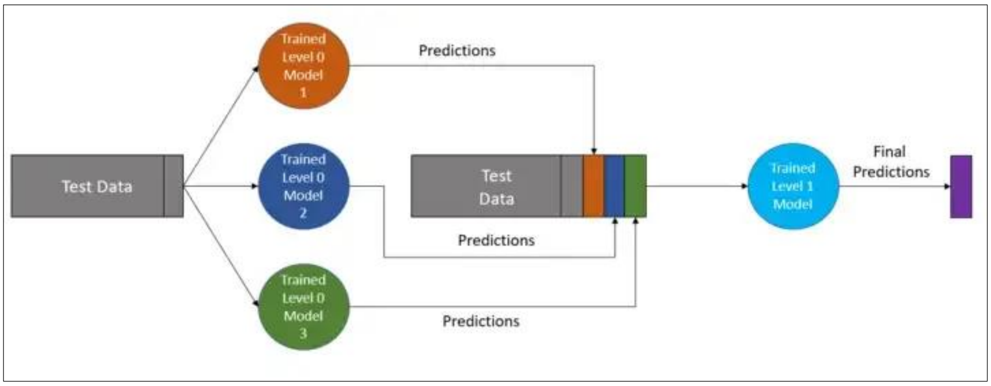


Figure 9. *Final Stacking Stage (Harrison, 2022).*

3. RESULTS AND DISCUSSION

The goal of this research was to implement machine learning (ML) algorithms to categorize the diabetes disease from a clinical dataset using the built-in scikit's learn algorithms. This presentation encompasses all the stages and procedures involved in understanding the data, preprocessing, training the algorithm to develop a model, and testing the model to evaluate classification outputs.

Exploratory Data Analysis Results

According to the EDA results, the dataset has fourteen attributes or columns and 753 records. Most of the data types are numerical—both floats and integers— but with two object or string data types that need to be converted to numerical data types to make them ideal for our algorithm to learn. Figures 12 and 13 give us further details of the percentage of missing values in the entire data. A total of 1.2% of data is missing from the dataset and the visualization shows that the majority of the data is not missing. However, the machine learning methods and Scikit Learn library used do not perform well when there are omissions in the data. Therefore, missing data may be replaced or removed out of the dataset.

The class or target variable distribution is shown in Figure 14 as well as the correlation between the feature variables and the target variable in Figure 15. It can be seen that there is a strong positive correlation between weight and body mass index, hip circumference, and waist circumference, as well as between the target variable and blood glucose level. However, there is also a strong negative correlation between height and the target variable. These observations show us that we have an idea of the feature variables that are important for predicting the target variable. With 656 negative outcomes and 97 positive results, the target variable is likewise imbalanced. This simply indicates that because of the enormous disparity in the frequency of the target class, if we utilize this dataset in this way, our models will learn more from the majority class (negative outcomes) and less from the minority class (positive outcomes). Hence, the dataset must be balanced.

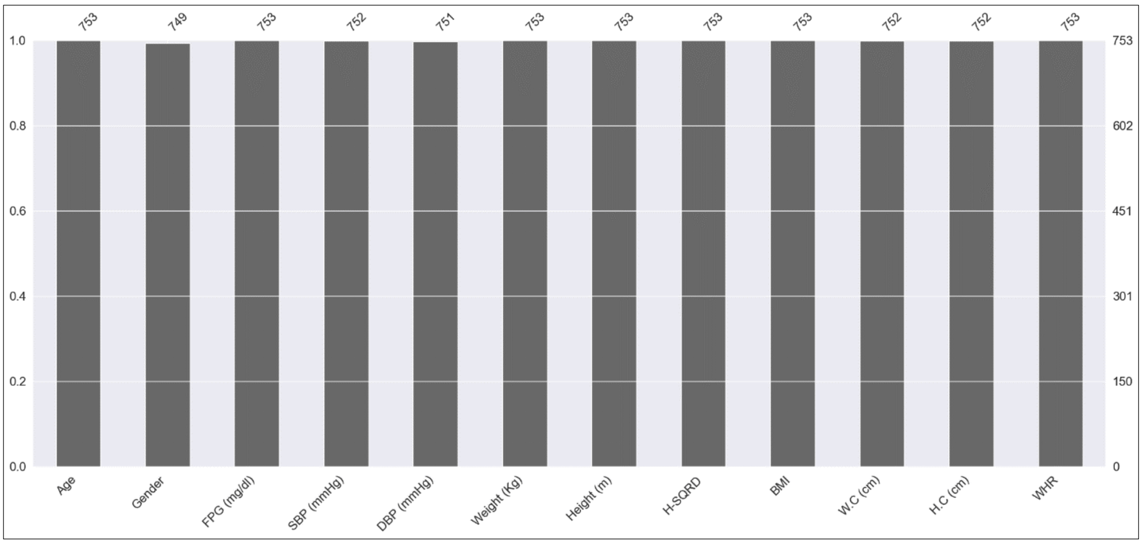
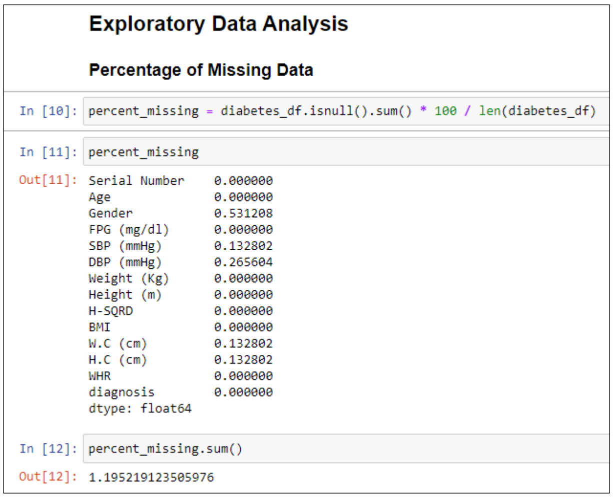


Figure 12. *Visualization of Missing Data in the Dataset*



Exploratory Data Analysis

Percentage of Missing Data

```
In [10]: percent_missing = diabetes_df.isnull().sum() * 100 / len(diabetes_df)

In [11]: percent_missing

Out[11]: Serial Number    0.000000
         Age              0.000000
         Gender           0.531208
         FPG (mg/dl)      0.000000
         SBP (mmHg)       0.132802
         DBP (mmHg)       0.265604
         Weight (Kg)      0.000000
         Height (m)       0.000000
         H-SQRD           0.000000
         BMI              0.000000
         W.C (cm)         0.132802
         H.C (cm)         0.132802
         WHR              0.000000
         diagnosis        0.000000
         dtype: float64

In [12]: percent_missing.sum()

Out[12]: 1.195219123505976
```

Figure 13. *Percentage of Missing Data in the Dataset*

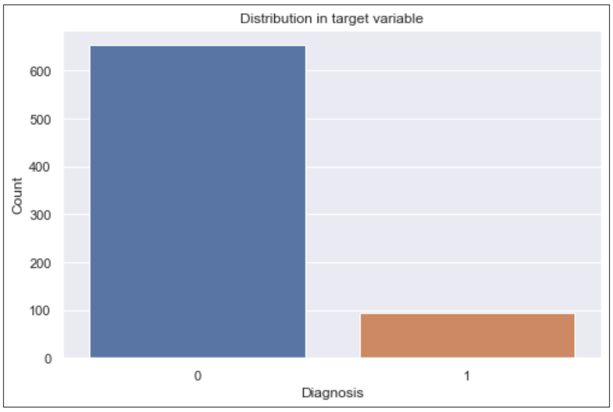


Figure 14. *Class Distribution of Target Variable*

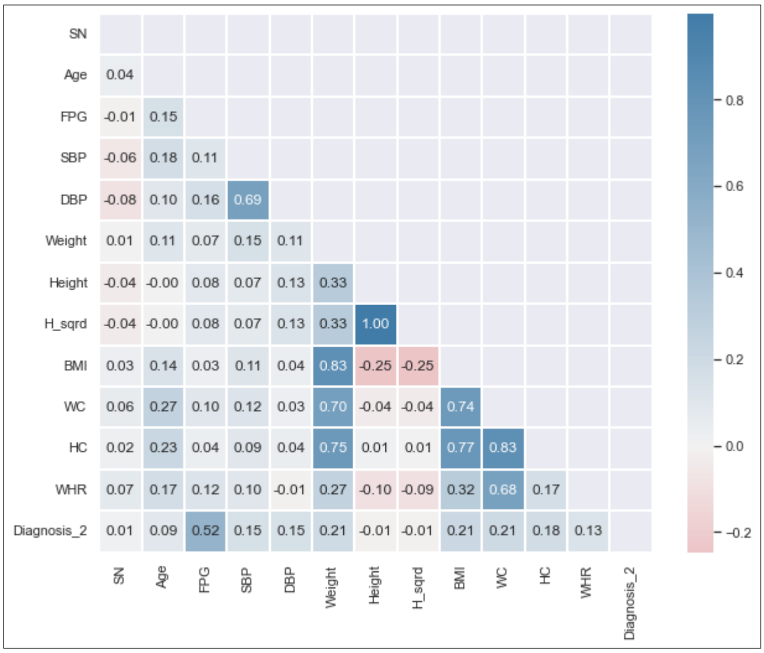


__Figure 15.__ Correlation between feature and target variable in the dataset

Data Preprocessing Results

We go one step further in this phase to fix the discrepancies we found in our dataset. Firstly, after identifying that only 1.2% present of our data is missing (that is 7 records out of 753), the records with the discovered missing data were dropped out of the dataset. This was done using Python's pandas dropna() method to remove all rows with missing data. The result of the dataset without missing values is shown in Figures 16 and 17.

Secondly, the dataset was balanced only on the training dataset after splitting the data into training and test sets. The SMOTE technique from the Imbalanced Learn Over Sampling library was used to balance our training dataset with both positive and negative classes having the same number of samples (588 each) as shown in Figure 18.

Thirdly, to standardize the data, feature scaling was also carried out on each feature variable from the training dataset using the standard scalar function of the scikit-learn preprocessing module. Finally, the most important features from the dataset were selected by using a recursive elimination technique from the Scikit Learn Feature Selection Library. After discovering these features, we used the PCA method from the Scikit Learn Decomposition Library, by specifying the number of important features earlier identified.

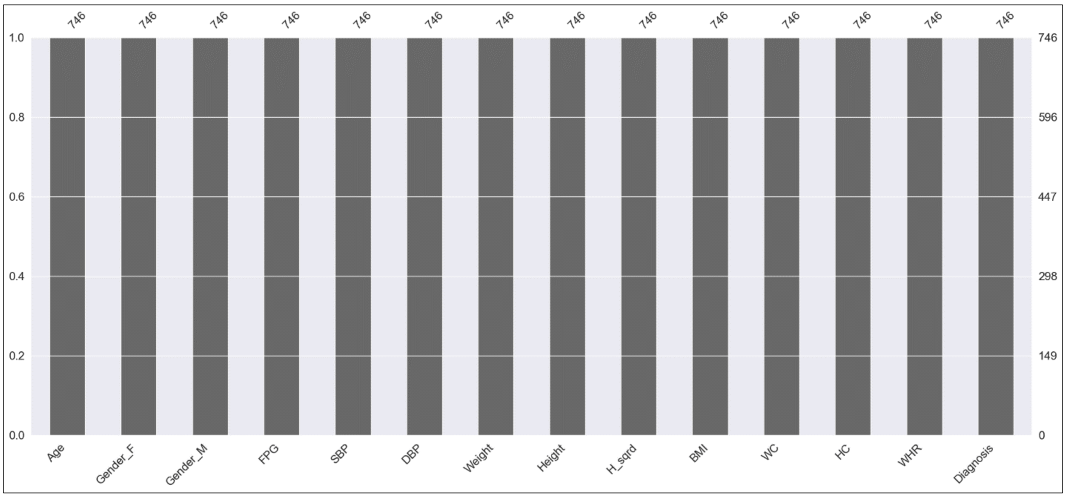
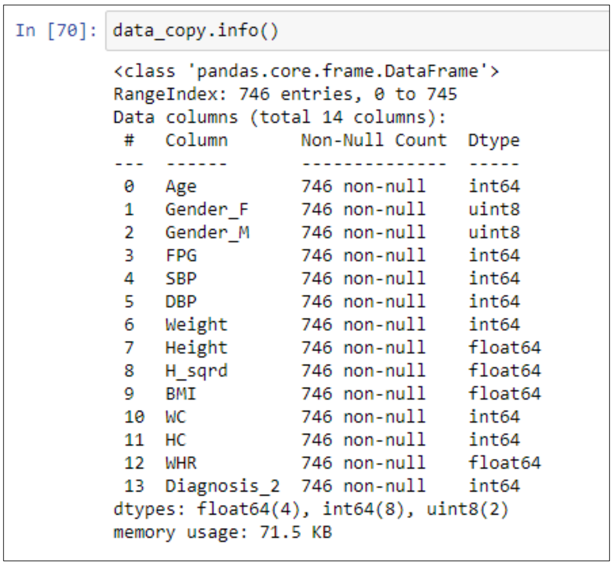


__Figure 16.__ Visualization of Dataset after Removing Missing Values.



```
In [70]: data_copy.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 746 entries, 0 to 745
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Age          746 non-null    int64
 1   Gender_F     746 non-null    uint8
 2   Gender_M     746 non-null    uint8
 3   FPG          746 non-null    int64
 4   SBP          746 non-null    int64
 5   DBP          746 non-null    int64
 6   Weight       746 non-null    int64
 7   Height       746 non-null    float64
 8   H_sqrd       746 non-null    float64
 9   BMI          746 non-null    float64
 10  WC           746 non-null    int64
 11  HC           746 non-null    int64
 12  WHR          746 non-null    float64
 13  Diagnosis_2  746 non-null    int64
dtypes: float64(4), int64(8), uint8(2)
memory usage: 71.5 KB
```

Figure 17. *Result after Removing Missing Values*

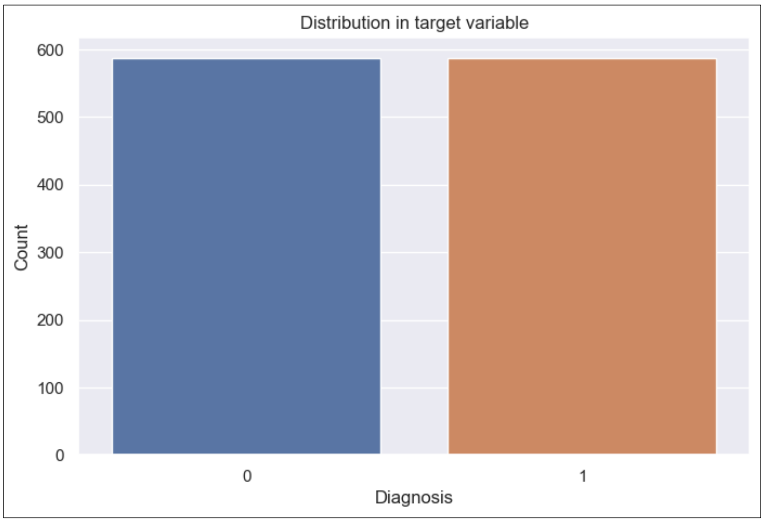


Figure 18. *Result after using SMOTE to Balance the Target Class on the Training Dataset.*

Setup for the Experiment

The essential experiments at different stages of our suggested technique were carried out in this study using various tools and frameworks. Conda, a powerful tool used to manage all other tools and frameworks, is the primary environment and package manager.

Python-based libraries utilized in the Conda environment include Pandas, Numpy, Matplotlib, and Scikit Learn. Jupyter notebook is an interactive web-based tool, also known as a computational notebook, that was used to build the machine learning algorithms and the libraries for data analysis, visualization, and machine learning stated above.

The Datacamp workspace, which includes an online Jupyter lab for data science and machine learning tasks, is another alternative and helpful web environment that was used for this research. The benefits of employing a cloud-based data science platform for machine learning tasks are demonstrated in this workspace. Many pre-installed modules are accessible for usage, and one may access his or her work remotely from any device without worrying about dependency difficulties.

After preprocessing our data, we performed several experiments utilizing all five machine-learning techniques. We divided our data into training and testing sets, using the Stratified K-fold Cross-validation technique from the Scikit Learn Model Selection Library. This method was used to split our dataset and to also perform a 10 Cross-Fold validation on the dataset. In addition to using this method, we specified a random state value of 529 to allow us to reproduce the results repeatedly. Finally, we sent a Numpy array of our preprocessed data to our algorithm.

Confusion Matrix Result

We successfully built a model from our training data and then utilized our testing data to assess how well each of the algorithms in this research performed. All performance matrices were computed from the confusion matrix for each model.

Figure 19 and 20 displays the findings for the confusion matrix for the Support Vector Machine and Stacked Ensemble Model. Each of its characteristics has a value of TP = 64, TN = 8, FP = 1, and FN = 1. The performance assessment metrics that produced the greatest outcomes in this study were computed using the values that the Support Vector Machine and Stacked Ensemble Model generated.

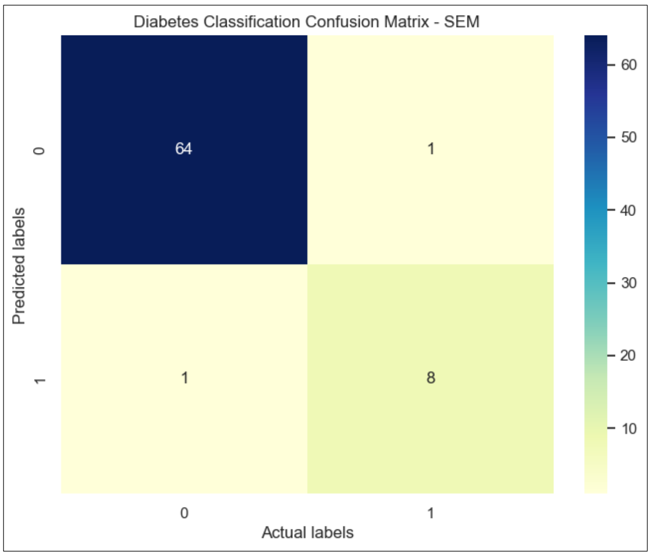


Figure 19. *Confusion Matrix for Diabetes Classification with SEM*

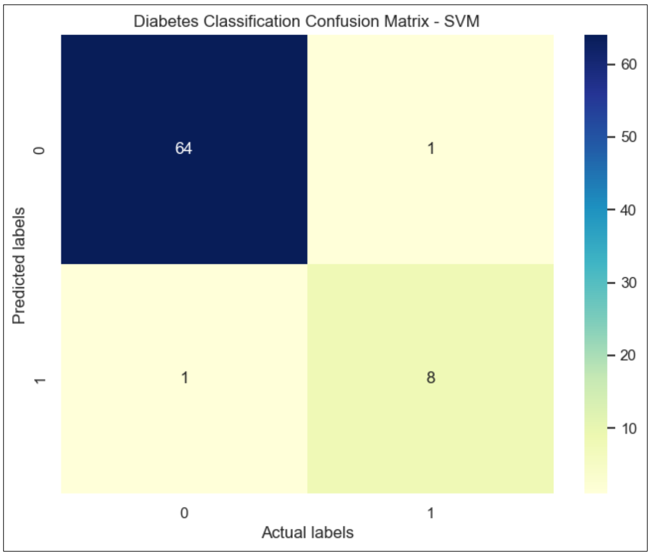
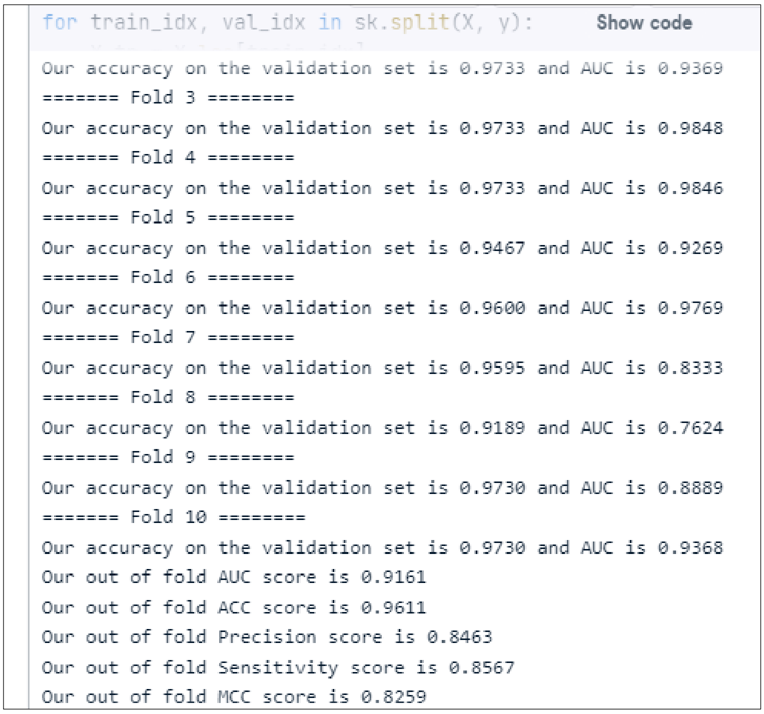


Figure 20. *Confusion Matrix for Diabetes Classification with SVM*

Evaluation Metrics Results

The values produced by the confusion matrices in Figures 19 and 20 were used to compute the output for the evaluation metrics in Figures 21, 22, 23, and 24. The outcome shows that, compared to all the other evaluation metrics employed in this study, the Accuracy and ROC AUC score assessment metrics had a more significant percentage. Furthermore, all results were computed using 10-fold cross-validation on all evaluation metrics. The final score for each measure is the mean of the total number of predictions. Cross-validation gives a more reliable result to check against overfitting. Finally, Figure 25 shows the graphical representation of the Stacked Ensemble Method model in a pipeline with Logistic Regression as the final estimator.



```
for train_idx, val_idx in sk.split(X, y):      Show code
Our accuracy on the validation set is 0.9733 and AUC is 0.9369
======= Fold 3 ========
Our accuracy on the validation set is 0.9733 and AUC is 0.9848
======= Fold 4 ========
Our accuracy on the validation set is 0.9733 and AUC is 0.9846
======= Fold 5 ========
Our accuracy on the validation set is 0.9467 and AUC is 0.9269
======= Fold 6 ========
Our accuracy on the validation set is 0.9600 and AUC is 0.9769
======= Fold 7 ========
Our accuracy on the validation set is 0.9595 and AUC is 0.8333
======= Fold 8 ========
Our accuracy on the validation set is 0.9189 and AUC is 0.7624
======= Fold 9 ========
Our accuracy on the validation set is 0.9730 and AUC is 0.8889
======= Fold 10 ========
Our accuracy on the validation set is 0.9730 and AUC is 0.9368
Our out of fold AUC score is 0.9161
Our out of fold ACC score is 0.9611
Our out of fold Precision score is 0.8463
Our out of fold Sensitivity score is 0.8567
Our out of fold MCC score is 0.8259
```

Figure 21. *Performance Evaluation Metrics Results for SVM*

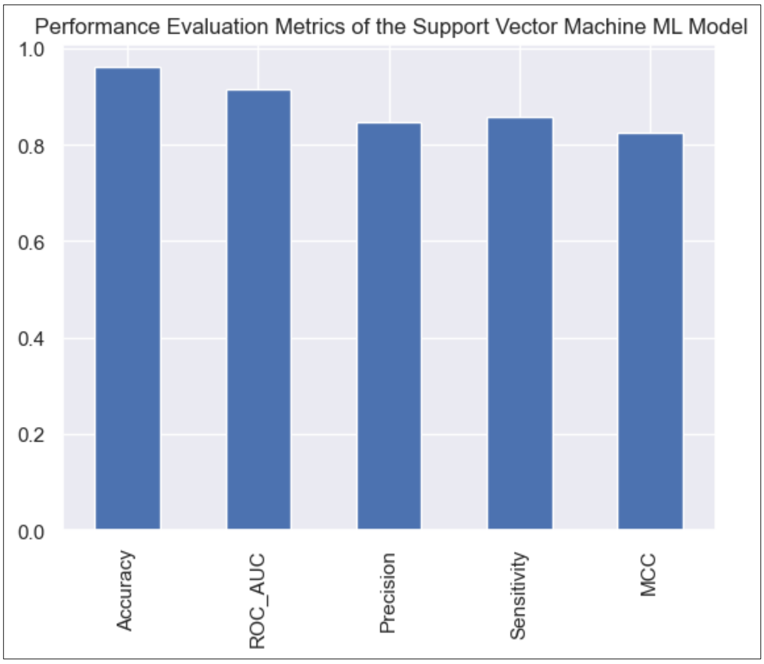
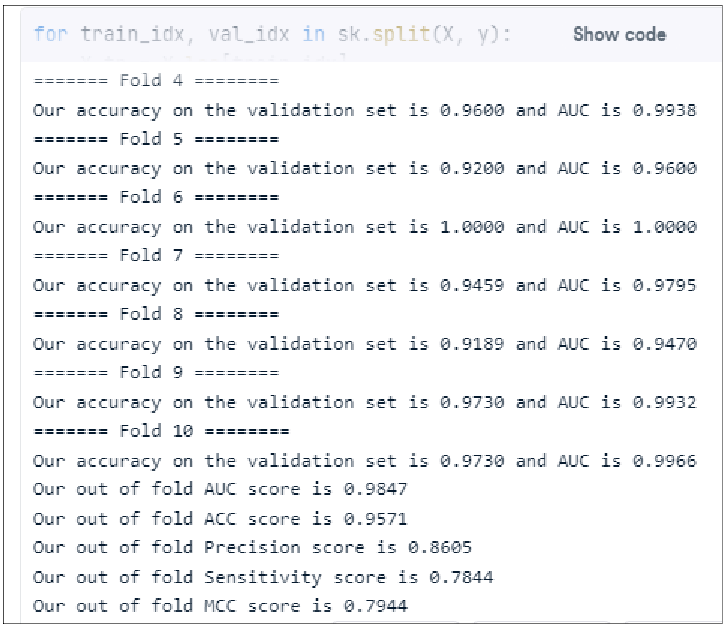


Figure 22. *Comparing all Five Evaluation Metrics for the Support Vector Machine ML Model*



```
for train_idx, val_idx in sk.split(X, y):
```

```
======= Fold 4 ========
Our accuracy on the validation set is 0.9600 and AUC is 0.9938
======= Fold 5 ========
Our accuracy on the validation set is 0.9200 and AUC is 0.9600
======= Fold 6 ========
Our accuracy on the validation set is 1.0000 and AUC is 1.0000
======= Fold 7 ========
Our accuracy on the validation set is 0.9459 and AUC is 0.9795
======= Fold 8 ========
Our accuracy on the validation set is 0.9189 and AUC is 0.9470
======= Fold 9 ========
Our accuracy on the validation set is 0.9730 and AUC is 0.9932
======= Fold 10 ========
Our accuracy on the validation set is 0.9730 and AUC is 0.9966
Our out of fold AUC score is 0.9847
Our out of fold ACC score is 0.9571
Our out of fold Precision score is 0.8605
Our out of fold Sensitivity score is 0.7844
Our out of fold MCC score is 0.7944
```

Figure 23. *Performance Evaluation Metrics Results for the Stacked Ensemble Model*

Figure 24 graph shows that the area under the receiver operating characteristic curve has a value range significantly closer to 1 than 0.5.

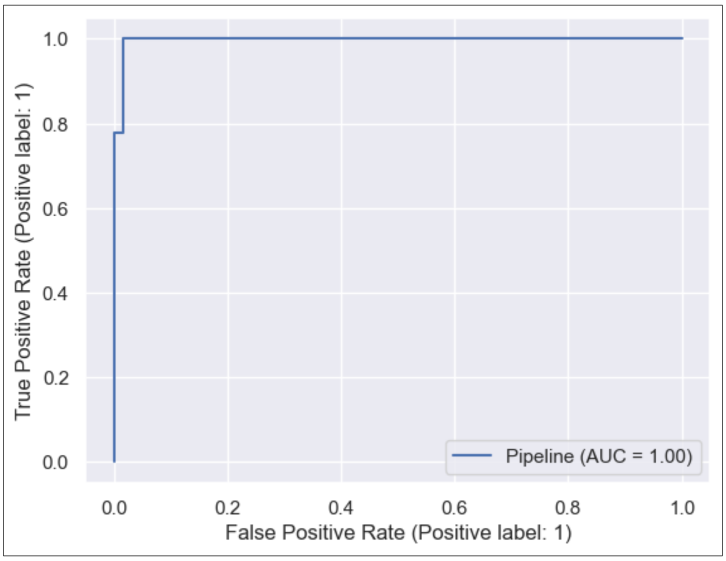


Figure 24. *ROC_AUC Graph for our SEM Model*

Comparison of the ML algorithms Employed in this Research

Table 2 compares the findings of the five evaluation metrics that were utilized in this study, and it shows that our Stacked Ensemble Method was the best model based on an ROC AUC score of **98.47 %**. However, SVM performed best in all other evaluation metrics with an Accuracy score of **96.11 %**, Precision score of **91.61%**, Sensitivity Score of **85.67%**, and Matthew's Correlation Coefficient score of **82.59 %**. The Stacked Ensemble Method model was the second best in all other evaluation metrics. Finally, Table 3 compares the findings of the combinations of different stacked models. The first combination was the stacking of the trained Support Vector Machine and Random Forest models (SEM_2). These were the two best algorithms from Table 2 without considering the stacked ensemble model column results reported. The next combination was the stacking of Support Vector Machine, Random Forest, and Logistic Regression models (SEM_3). Lastly, we stacked all five trained models (SEM_5) used in this study which produced the best results when compared to all other stacked ensemble methods employed.

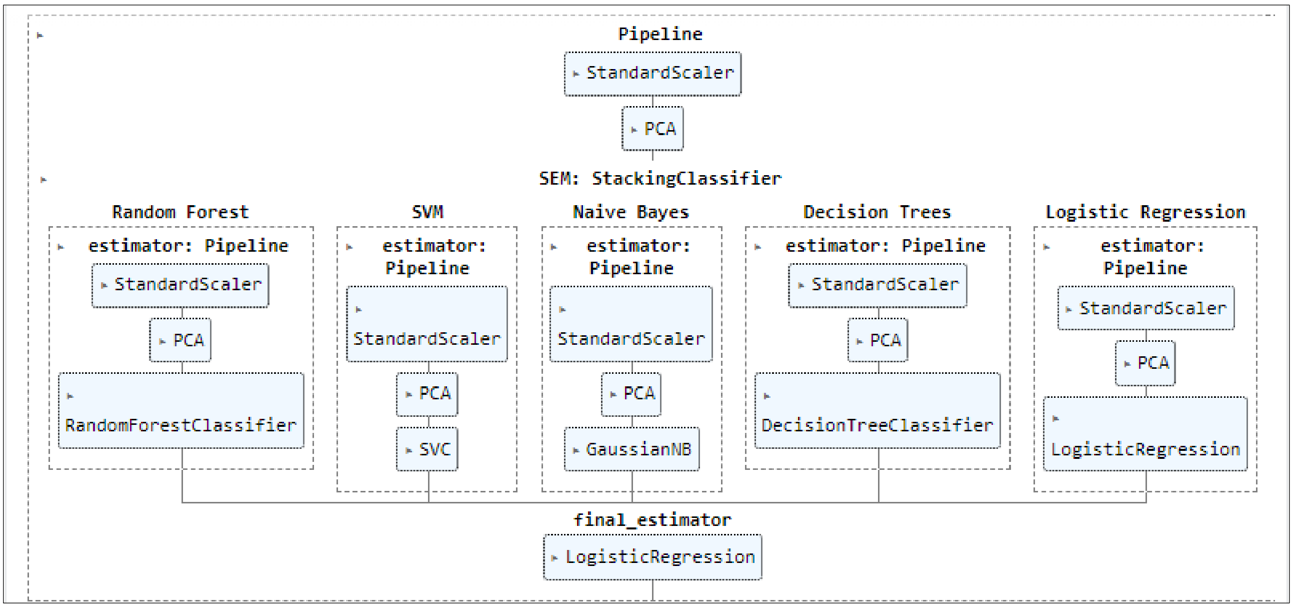


Figure 25. *SEM Model in a Pipeline*

Table 2. *Comparative Performance Analysis of the Five (5) ML Algorithm used in this Study*

Algorithm /Evaluation Metrics	Random Forest	Naïve Bayes	Logistic Regression	Support Vector Machine	Decision Tree	Stacked Ensemble Method
Accuracy	93.97 %	89.09 %	92.23 %	**96.11 %**	86.60 %	95.71 %
Precision	79.02 %	56.03 %	64.23 %	**91.61 %**	48.17 %	86.05 %
Sensitivity	70.44 %	67.44 %	84.78 %	**85.67 %**	66.11 %	78.44 %
MCC	70.66 %	54.83 %	69.51 %	**82.59 %**	48.84 %	79.44 %
AUC_ROC	96.06 %	89.86 %	96.02 %	91.53 %	85.41 %	**98.47 %**

Table 1. *Comparative Analysis of the Three (3) Different Stacked Ensemble Method Combinations.*

Stacked Models /Evaluation Metrics	Stacked Ensemble Method (SEM_2)	Stacked Ensemble Method (SEM_3)	Stacked Ensemble Method (SEM_5)
Accuracy	94.63 %	94.77 %	**95.71 %**
Precision	79.95 %	81.13 %	**86.05 %**
Sensitivity	75.00 %	76.22 %	**78.44 %**
MCC	74.16 %	75.23 %	**79.44 %**
AUC_ROC	97.68 %	97.86 %	**98.47 %**

4. CONCLUSION

To optimize the insights found to enhance better and more significant outcomes when used in the medical domain, more study is being conducted on the use of machine learning in detecting illnesses, notably diabetes. In this study, examining evaluation metrics other than accuracy alone and comparing them to other metrics yields excellent findings in terms of precision, sensitivity, and area under the receiver operating characteristic curve. These results were obtained by applying data preprocessing techniques since clinical data might be prone to missing data and discrepancies, in addition to understanding the working principles of the algorithms utilized.

Recommendation for Future Research

The classification approaches to predict diabetic disease produced acceptable accuracy, sensitivity, precision, MCC, and ROC AUC values. The following enhancements are recommended for this study's future research:

1. To enhance all measures employed in this study and utilize additional cutting-edge supervised machine learning and deep learning techniques.
2. A web-based or mobile application for diagnosing diabetic illness can use the best models as its backend.
3. To use clinical data with machine learning algorithms to determine the kind of diabetes a patient has (Type I, Type II, or Gestational Diabetes, etc.).

AUTHOR CONTRIBUTIONS

Conceptualization, Nathan Zoakah, Augustine Nsang and Abel Ajibesin; Methodology, Nathan Zoakah and Augustine Nsang; Data Source, Ayuba Zoakah; Software, Nathan Zoakah; Title, Augustine Nsang and Abel Ajibesin; Experiments: Nathan Zoakah, Augustine Nsang: Validation, Augustine Nsang, Ayuba Zoakah; Manuscript-original draft, Nathan Zoakah; Manuscript-review and editing, Abel Ajibesin, Ayuba Zoakah Augustine Nsang; Visualization, Nathan Zoakah; Supervision, Agustine Nsang, Ayuba Zoakah

ACKNOWLEDGEMENT

This study contains clinical data from The Apex Clinic and Diabetes Screening Centre in Jos, Plateau State, Nigeria. Their contribution to the success of this study is gratefully acknowledged.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Armstrong, A. (2022, March 1). Python in Healthcare: AI Applications in Hospitals. https://www.datacamp.com/blog/python-in-healthcare-ai-applications-in-hospitals?utm_medium=email&utm_source=customerio&utm_id=7430059&utm_campaign=dc_insights&utm_term=v2blog

Bhatia, P. (2019). *Data mining and data warehousing: Principles and practical techniques*. Cambridge University Press.

Birjais, R., Mourya, A. K., Chauhan, R., & Kaur, H. (2019). Prediction and diagnosis of future diabetes risk: A machine learning approach. *SN Applied Sciences*, *1*(9), 1112. https://doi.org/10.1007/s42452-019-1117-9

Choudhary, D. (2021, April 18). Bootstrapping and OOB samples in Random Forests. Analytics Vidhya. https://medium.com/analytics-vidhya/bootstrapping-and-oob-samples-in-random-forests-6e083b6bc341

Choudhury, A., & Gupta, D. (2019). *A Survey on Medical Diagnosis of Diabetes Using Machine Learning Techniques*. In: J. Kalita, V. E. Balas, S. Borah, & R. Pradhan (Eds.), Recent Developments in Machine Learning and Data Analytics (Vol. 740, pp. 67-78). Springer Singapore. https://doi.org/10.1007/978-981-13-1280-9_6

Gandhi, R. (2018, May 17). Naive Bayes Classifier. Medium. https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

Harrison, G. (2022, February 28). A Deep Dive into Stacking Ensemble Machine Learning—Part I. Medium. https://towardsdatascience.com/a-deep-dive-into-stacking-ensemble-machine-learning-part-i-10476b2ade3

Ibrahim, I., & Abdulazeez, A. (2021). The Role of Machine Learning Algorithms for Diagnosing Diseases. *Journal of Applied Science and Technology Trends*, *2*(01), 10-19. https://doi.org/10.38094/jastt20179

IDF (International Diabetes Federation) (2021). IDF Diabetes Atlas 10th ed.

Jakkula, V. (2010) Tutorial on Support Vector Machine (SVM).

Khanam, J. J., & Foo, S. Y. (2021). A comparison of machine learning algorithms for diabetes prediction. *ICT Express*, *7*(4), 432-439. https://doi.org/10.1016/j.icte.2021.02.004

Lanhenke, M. (2022, May 1). Implementing Support Vector Machine From Scratch. Medium. https://towardsdatascience.com/implementing-svm-from-scratch-784e4ad0bc6a

Loeber, P. (2019a, September 29). Naive Bayes in Python—Machine Learning From Scratch 05—Python Tutorial—YouTube. https://www.youtube.com/watch?v=BqUmKsfSWho&list=PLqnslRFeH2Upcrywf-u2etjdxxkL8nl7E&index=5

Loeber, P. (2019b, November 22). Decision Tree in Python Part 2/2—Machine Learning From Scratch 09—Python Tutorial. https://www.youtube.com/watch?v=Bqi7EFFvNOg

Loeber, P. (2019c, November 27). Random Forest in Python—Machine Learning From Scratch 10—Python Tutorial. https://www.youtube.com/watch?v=Oq1cKjR8hNo

Maniruzzaman, Md., Rahman, Md. J., Ahammed, B., & Abedin, Md. M. (2020). Classification and prediction of diabetes disease using machine learning paradigm. *Health Information Science and Systems*, *8*(1), 7. https://doi.org/10.1007/s13755-019-0095-z

Normalized Nerd (Director). (2021, January 13). Decision Tree Classification Clearly Explained! https://www.youtube.com/watch?v=ZVR2Way4nwQ

Pranto, B., Mehnaz, S., Mahid, E. B., Sadman, I. M., Rahman, A., & Momen, S. (2020). Evaluating machine learning methods for predicting diabetes among female patients in Bangladesh. *Information*, *11*(8), 374. https://doi.org/10.3390/info11080374

Prasanna, S. (2019). *Machine Learning with Python*. 1, 167.

Punthakee, Z., Goldenberg, R., & Katz, P. (2018). Definition, Classification and Diagnosis of Diabetes, Prediabetes and Metabolic Syndrome. *Canadian Journal of Diabetes*, *42*, S10-S15. https://doi.org/10.1016/j.jcjd.2017.10.003

Rokach, L. (2009). Pattern Classification Using Ensemble Methods (Illustrated edition, Vol. 75). World Scientific Publishing Company.

Shmueli, G., Bruce, P. C., Gedeck, P., & Patel, N. R. (2019). *Data mining for business analytics: Concepts, techniques and applications in Python* (1st ed.). John Wiley & Sons.

Singh, H. (2021, March 30). Variants of Stacking | Types of Stacking—Advanced Ensemble Learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/03/advanced-ensemble-learning-technique-stacking-and-its-variants/

Sisodia, D., & Sisodia, D. S. (2018). Prediction of diabetes using classification algorithms. *Procedia Computer Science*, *132*, 1578-1585.

Sruthi, E. R. (2021, June 17). Random Forest | Introduction to Random Forest Algorithm. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

Swaminathan, S. (2019, January 18). Logistic Regression—Detailed Overview. Medium. https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

Tigga, N. P., & Garg, S. (2020). Prediction of type 2 diabetes using machine learning classification methods. *Procedia Computer Science*, *167*, 706-716.

WHO (World Health Organization) (2021, November 10). Diabetes. https://www.who.int/news-room/fact-sheets/detail/diabetes

Zhang, C., & Ma, Y. (Eds.). (2012). *Ensemble Machine Learning: Methods and Applications* (2012th edition). Springer.

Zhu, C., Idemudia, C. U., & Feng, W. (2019). Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques. *Informatics in Medicine Unlocked*, *17*, 100179.

Zimmet, P., Alberti, K. G., Magliano, D. J., & Bennett, P. H. (2016). Diabetes mellitus statistics on prevalence and mortality: Facts and fallacies. *Nature Reviews Endocrinology*, *12*(10), 616-622.