Journal of The Faculty of Computational Sciences & Informatics
Academic City University College Accra, Ghana
Vol 4 No 1, March 2024

Research Article

Journal of Computational Sciences & Informatics

A Logistic Regression-Based Technique for Predicting Type II Diabetes

[1]Babatunde Ronke Seyi, [2]Babatunde Akinbowale Nathaniel, [3]Balogun Bukola Fatimah, [4]AbdulRahman Tosho Abdulahi, [5]Umar Emmanuel, [6]Ajiboye Raimot Adenike, [7]Mohammed Shuaib Babatunde, [8]Oke, Afeez Adeshina & [9]Obiwusi, Kolawole Yusuf
[1, 2, 3, 7]Kwara State University, Malete, PMB 1530, Ilorin, Kwara State, Nigeria;
ronke.babatunde@kwasu.edu.ng, akinbowale.babatunde@kwasu.edu.ng*,
bukola.balogun@kwasu.edu.ng, shuiab.mohammed@kwasu.edu.ng,
[4,6] Kwara State Polytechnic, Ilorin, Kwara State, Nigeria;
abdulrahman.t@kwarastatepolytechnic.edu.ng, ajiboye.r@kwarastatepolytechnic.edu.ng,
[5]Tai Solarin University of Education, Ogun state; umaremmanuel5@gmail.com,
[8]Federal College of Education, Iwo, Osun State; okeaa@fceiwo.edu.ng
[9]Summit University, Offa, Kwara State, Nigeria; obiwusi.kolawole@summituniversity.edu.ng.

ABSTRACT

In recent years, diabetes has emerged as one of the main causes of death for people. The spread of unhealthy foods, sedentary lifestyles, and eating habits have all contributed to the annual increase in the incidence of diabetes. A diabetes prediction model can help with clinical management decision-making. Diabetes prevention may be aided by being aware of potential risk factors and early detection of high-risk individuals. Numerous diabetes prediction models have been created. The size of the data set to be used was an issue in earlier research, but more recent studies have incorporated the use of high-quality, trustworthy data sets, such as the Vanderbilt and PIMA India data sets. Recent research has demonstrated that a few variables, including glucose, pregnancy, body mass index (BMI), the function of the diabetic pedigree, and age, can be used to predict Type II diabetes. Machine learning models of these parameters can be used to accurately predict the chance of the disease occurring as it was investigated in this study. In order to predict Type II diabetes, this study used the machine learning method Logistic Regression.

Keywords: Type II Diabetes, Logistic Regression, Hyperparameter and Prediction

ACity FCSI Journal Citation Format
Babatunde, R.S., Babatunde, A.N., Balogun, B.F., Abdulrahman, T.A., Umar, E., Ajiboye, R.A., Mohammed, S.B., Oke, A.A. & Obiwusi, K.Y. (2024): A Logistic Regression-Based Technique for Predicting Type II Diabetes. Journal of Computational Sciences & Informatics. Academic City University College, Accra, Ghana. Vol 4 No. 1 March, 2024 Pp 1-14. https://www.isteams.net/ghanabespoke2023. dx.doi.org/10.22624/AIMS/FCSIJ/2024/P1

1. BACKGROUND TO THE STUD

One of the most common and urgent public health issues in the world today is diabetes. The prevalence of type 2 diabetes has been rising worldwide. Diabetes lowers quality of life and promotes early mortality; it is linked to numerous other disorders, including those of the kidney, heart, lower limb, and eye [1, 13].

Its importance is highlighted by the startling data, which show that 1.37 million people worldwide died from diabetes in 2017 and almost 450 million individuals were diagnosed with the disease [2]. Over 100 million persons in the US suffer with diabetes, and by 2020, it ranked as the seventh most common cause of death in the nation [3]. The consequences are severe since diabetes increases the risk of serious health issues such as kidney failure, heart disease, stroke, early death, and amputation of limbs, which frequently results in tissue damage and long-term disability [4, 13]. The effects of diabetes go beyond the immediate expenses and include loss and unproductivity at work.

Interestingly, people with a high risk of diabetes might not be aware of the risk factors that go along with it. Because diabetes is so common and severe, scientists are interested in finding the risk factors that lead to the disease's development. Early risk factor identification and diabetes prediction are essential for reducing the financial burden associated with diabetes, preventing complications from the disease and providing significant advantages from the standpoints of public health and clinical practice [3]. Informed clinical care is aided by prediction models, which screen for pre-diabetes or people with an increased risk of acquiring diabetes. To assess the risk factors linked to incident of diabetes, a number of prediction models have been put out [5]. Furthermore, [6] used deep learning algorithms to forecast when diabetes will manifest, suggesting the possibility of advanced techniques to improve model performance. On the other hand, a number of studies have indicated that for predicting the risk of an illness, logistic regression works just as well as machine learning methods.

While many different predictive models have been used to forecast Type II diabetes, insufficient covariate selection, missing data, limited sample sizes, and incorrectly stated statistical models have all contributed to the lack of strong model development. Notably, [5] discovered that the logistic regression model produced better accuracy when used in conjunction with machine learning methods. The evaluation of diabetes risk factors, such as household characteristics, forms the basis of these models.

Improving patient outcomes depends on Type II diabetes early detection and prevention. There is still a need for a reliable Type II diabetes risk prediction model, despite improvements in diagnostic methods. However, owing to the fact that Type II diabetes is fatal, higher death rates have resulted from the lack of accurate predictive model [6]. The goal of this research work is to create a Type II diabetes prediction model using logistic regression by considering the effect of feature size on the predictive power of the model. The objectives include gathering diabetes data, validating the entire dataset, selecting features that are most important for predicting Type II diabetes, training the model, fine-tuning parameters while the model is in use, and assessing the model's effectiveness based on prediction accuracy.

The Type II diabetes risk prediction model in this research uses algorithms and methodologies based on Logistic Regression and other Machine Learning approaches. An approach for classification called logistic regression uses a set of independent factors to predict binary outcomes. In order to evaluate accuracy and error rates, this study takes validation and cross-validation mistakes into account. This study raises awareness of Type II Diabetes symptoms and their long-term effects. It also clarifies the several variables that might be employed in the identification and diagnosis of individuals suffering from Type II Diabetes. Furthermore, this research is essential in lowering the high death rate from Type II Diabetes. This discovery helps to lower the death rate linked to Type II Diabetes by enabling a quicker and more effective diagnosis.

2. RELATED WORK

This section offers an insightful overview of various research publications and articles focusing on the assessment and prediction of Type 2 Diabetes Mellitus (T2DM) risk. The objective of these studies is to enhance the precision and reliability of T2DM prediction and early detection. Each of these investigations employs diverse machine learning methodologies, datasets, and performance evaluation metrics. It's important to bear in mind that, like any scientific research, these studies have their own set of limitations. [2] introduces a data mining-based method for T2DM prediction, utilizing improved Logistic Regression with Multilayer Perception (LRMLP) and Naïve Bayes algorithms with preprocessing techniques. However, the study falls short in including key parameters like specificity and the area under the receiver operating characteristic curve (AUC-ROC), which are essential for a comprehensive assessment of the model's performance. [1] develop a system to predict diabetes risk variables by employing machine learning techniques.

They incorporate feature selection methods such as PCA and IG to enhance prediction accuracy, achieving an AUC value of 87.2%, surpassing an 82.2% accuracy rate. Nonetheless, the study suggests that further exploration of sophisticated feature selection and optimization strategies could yield even better results. [7] developed a Type 2 Diabetes Mellitus Prediction model using Logistic Regression based scorecards. They employed gradient boosting decision trees and logistic regression to forecast the likelihood that type 2 diabetes would manifest within a given time range. The study achieved an auROC of 0.81 using the auROC as a performance parameter. The lack of comprehensive insights into the operation of gradient boosting decision trees and possible improvements is a weakness of the study. Logistic regression was used in the risk prediction model proposed by [8, 14]. The Ganzhou city's 19 districts and counties provided the dataset for the baseline survey. In terms of females, the model's sensitivity was 0.75 and its specificity was 0.90. The study exclusively focuses on the performance indicators of particular gender groups and lack of investigation into more general applications is one of its limitations.

[9] used logistic regression and ensemble techniques for the development of a diabetes predictive model. T2DM was predicted using logistic regression in addition to other machine learning algorithms. The PIMA India dataset and the Vanderbilt dataset were used in the analysis. Using ensemble approaches resulted in the maximum accuracy of about 78% for Dataset 1 and almost 93% for Dataset 2. The study's failure to look at new algorithms or feature engineering techniques for possible model improvement is one of its limitations. Using the PIMA Indian dataset, [10] presented a risk prediction model for the Prediction of Type 2 Diabetes Using Logistic Regression. Based on their investigations, the decision tree (c5.0) had the best accuracy of all the models that were taken into consideration with an accuracy of 80.68%. Other sophisticated modeling methods that can increase prediction accuracy were not examined in the study.

[11] presents the Average Weighted Objective Distance (AWOD) method, a novel prediction technique for T2DM. This approach adjusts the Weighted Objective Distance (WOD) by considering individual traits and different priorities. However, the mechanics of the AWOD method are not explained in great detail, which can pose challenges for readers seeking a deeper understanding. [12] developed a risk prediction model for T2DM using the PIMA Indian dataset, employing decision trees and logistic regression models. The study identifies crucial T2DM variables, including glucose, BMI, pregnancy, the function of the diabetes pedigree, and age. Nevertheless, the study's focus on basic model enhancements may limit prediction accuracy.

While extensive work is ongoing in this domain, majority of the previous studies either lack in-depth evaluations of the effectiveness of various machine learning algorithms beyond accuracy or they lack sophisticated tools and methodologies in their development, hence the need for our proposed work. Our research aims to bridge these gaps by designing a more robust methodology and incorporating a comprehensive set of metrics, including accuracy, sensitivity, ROC AUC, and F1 Score, thus providing a thorough evaluation of the effectiveness of various machine learning algorithms. The rest of this paper is arranged as follows; section 2 discussed related research work while section 3 presents the materials and methods. Section 4 explains the experimental setup and section 5 discussed the results of the experiment. The conclusion and recommendation is presented in section 6.

3. MATERIALS AND METHODS

To achieve the project's objectives, a risk prediction model for Type II Diabetes will be developed using Binary logistic regression and other machine learning techniques. The methodology is structured into several steps: Data collection, Data processing, Feature selection, Model development, Model evaluation, and Model deployment. The tools used for this study include Python IDE, R Studio, and the Waikato environment for knowledge analysis toolkit, which are essential for data analysis and model development.

3.1 Data Acquisition

The first step in this methodology is collecting the required data for the research. The dataset includes clinical data of patients diagnosed with Type 2 diabetes, encompassing information such as age, symptoms, medical history, and other relevant data. For this study, a dataset from Kaggle was chosen. It includes attributes like glucose, pregnancy, body mass index (BMI), diabetes pedigree function, and age related to Type II Diabetes. Specifically, the PIMA Indian dataset was selected from Kaggle, which contains records of 7658 individuals diagnosed with Type 2 Diabetes. Among these, 500 individuals were labeled as "1" (indicating they have diabetes), while 268 were labeled as "0" (indicating they do not have diabetes).

3.2 Data Preprocessing and Cleaning

Data preprocessing and cleaning involve identifying and rectifying errors, inconsistencies, and inaccuracies in the raw data. The used dataset was preprocessed and cleaned to avoid missing values, repetitions and inaccuracies.

3.3 Feature Selection

Feature selection aims to improve the predictive accuracy and effectiveness of the system. It involves determining the most important features using domain expertise and statistical methods, including correlation analysis and feature importance derived from the gradient boosting algorithm. Feature selection helps reduce dimensionality.

3.4 Model Training

Once data is preprocessed and relevant features are selected, the next step is model training. This will be accomplished using Binary Logistic Regression, a classification model that uses probabilistic estimations to understand the relationship between the dependent and independent variables. A portion of the dataset is allocated as a training set, with the remainder reserved for testing.

3.5 Hyper-Parameter Tuning

After model training, hyper-parameter tuning is performed to optimize model performance. This process ensures that the model provides accurate results. Hyper-parameters are external configuration variables used to manage machine learning model training.

3.6 Model Evaluation

Following model training and hyper-parameter tuning, the final step is model evaluation. Various evaluation metrics, including accuracy and cross-validation error rates, are employed to assess the model's performance. Validation and cross-validation methods are used to validate the model using a dataset of Type 2 diabetes patients.

3.7 Classification using Logistic Regression

Logistic regression is a classification model commonly used in clinical analysis. It uses probabilistic estimates to understand the relationship between the dependent variable and independent variables. In the context of this study, the classification task results in binary outcomes, with "Type 1" indicating the absence of diabetes and "Type 0" indicating the presence of the disease. The proposed methodology follows a sequential flow, as depicted in Figure 1. This flowchart illustrates the progression from data collection to data processing, feature selection, model development, parameter tuning, model evaluation, and ultimately model utilization.

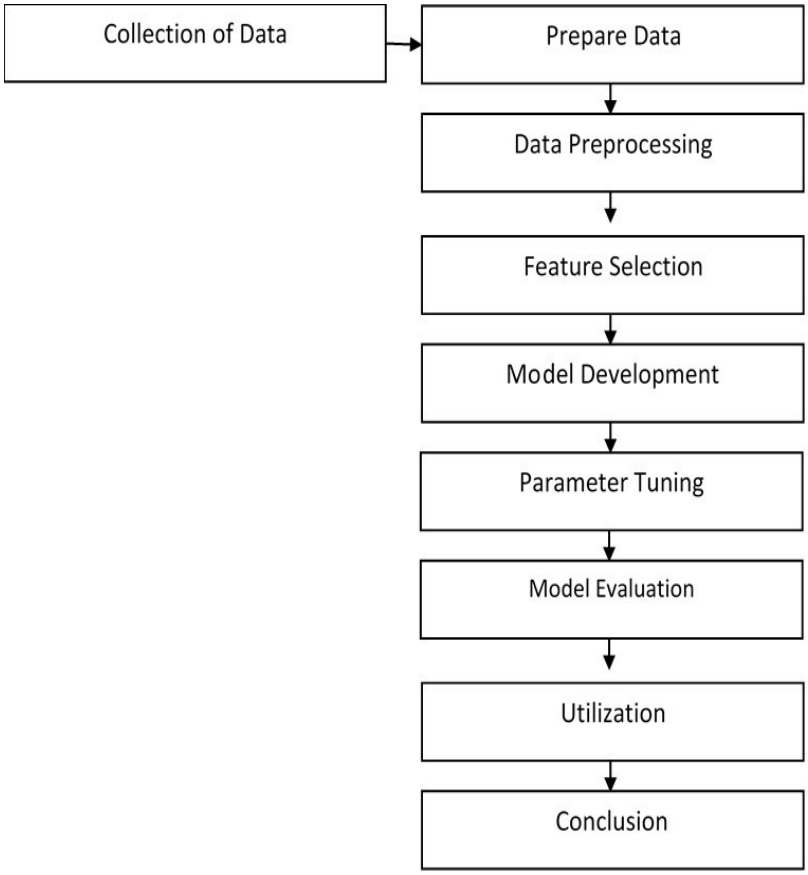

Figure 1: Process flow of the proposed system

4. EXPERIMENTAL PROCEDURE

After downloading the PIMA Indian dataset and importing the dataset, the next thing is the Data pre-processing. The Data pre-processing involves importing the dataset into the tool, cross checking and cleaning the data to remove any missing or inconsistent values and variables. Normalizing the data to ensure that all the features have same scale and style, checking for null values and transforming the data to make it suitable for machine learning. Figure 2 below shows how the dataset was imported using panda read excel function. It also displayed the first 10 rows of the imported data set.

Out[10]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

Figure 2: Dataset Importation and Visualization

Accessing and importing the dataset into the project is what is meant by dataset importation. The dataset can be kept in a number of different formats, including CSV, Excel, and SQL files. Depending on the dataset format and the programming language being used, an import technique can be chosen. Due to its effective data manipulation features, the Pandas package is frequently used for dataset importing in Python programs. It offers read_csv() and read_excel() utilities that make it simple to load datasets into a pandas DataFrame, a flexible data structure for processing tabular data. The jupyter interface was used to import the research's obtained dataset. The first five rows of the dataset were clearly displayed in Figure 3 below shows how the dataset was properly checked for null values and some inconsistent values

```
In [4]:  df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Figure 3: Checking for null and inconsistent values in the dataset

The database was cross checked to detect if there are any null set and at the end of the check there were no null values at all in the database. The data was already in a suitable condition for machine learning as it is well arranged and has no missing values in it.

4.2 Selection of Features

The feature selection method used for this system implementation was carried out by Selecting "The most significant features'' in the dataset. This is done in order to reduce the bulkiness and irrelevant complexity of the model. In this case, the PIMA India dataset has 8 features but in order to make things easier and faster, only 4 out of these features will be selected first based on their level of significance and importance. As the implementation goes on and in order not to doubt and question the reliability of the first 4 features selected, 2 other features were added to make it six (6) after they have been implemented to make sure the model and the features are reliable. Figure 4 below shows the selection of Four Features from the dataset based on the most significant features

	Pregnancies	Glucose	BMI	Age
0	6	148	33.6	50
1	1	85	26.6	31
2	8	183	23.3	32
3	1	89	28.1	21
4	0	137	43.1	33
5	5	116	25.6	30
6	3	78	31.0	26
7	10	115	35.3	29
8	2	197	30.5	53
9	8	125	0.0	54

Figure 4: Selection of Four Features from the dataset

As shown in Figure 4, four out of the 8 features were first selected based on their level of significance and importance. The first four features that were selected are Pregnancy, Glucose, BMI, and Age. These four features are considered the most significant features of the data set because they are the main factors that pre-determines if a patient has diabetes or not. Their values are indicated under it following a serial no of 0 to 9. Figure 5 below shows the selection of Four Features from the dataset based on the most significant features.

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	Age
0	6	148	72	0	33.6	50
1	1	85	66	0	26.6	31
2	8	183	64	0	23.3	32
3	1	89	66	94	28.1	21
4	0	137	40	168	43.1	33
5	5	116	74	0	25.6	30
6	3	78	50	88	31.0	26
7	10	115	0	0	35.3	29
8	2	197	70	543	30.5	53
9	8	125	96	0	0.0	54

Figure 5: Selection of six Features from the dataset

As shown in Figure 5, six (6) out of the 8 features were first selected based on their level of significance and importance. The two (2) added features were Insulin and Blood pressure. The values of these 6 features are indicated under it following a serial no of 0 to 9.

4.3 Training the model

Partitioning the data into training and testing sets is a crucial step in building a machine learning model, following preprocessing and feature selection. This allows one to assess the model's performance on untested data in addition to testing the model's generalization abilities. The "train-test split" technique was used to separate the dataset used in this study into training and testing sets. When the train-test split is employed, a random portion of the dataset is used as the testing set and the remaining part as the training set. A portion of the dataset is allocated to the testing set, contingent on the size of the dataset and the project's specifications. In this research work, 30% of the dataset was set aside for testing and the remaining 70% was used to train the model. Overfitting, which occurs when a model performs exceptionally well on training data but is unable to generalize to new, uncontaminated data, is something that the train-test split is crucial in preventing. By evaluating the model's performance on the testing set, we can predict how well it will perform on new data. The dataset is divided into two for each of the four features, as shown in Figure 6.

	Pregnancies	Glucose	BMI	Age
414	0	138	34.6	21
248	9	124	35.4	34
71	5	139	28.6	26
357	13	129	39.9	44
449	0	120	30.5	26
...	...	...	...	...
26	7	147	39.4	43
473	7	136	29.9	50
8	2	197	30.5	53
400	4	95	32.0	31
489	8	194	26.1	67

Figure 6: Data splitting for Four (4) Features

The chart above shows the data for the first four features divided. Using a method called "train-test split," the dataset for the four features were separated into training and testing sets. When using the train-test split, a random subset of the dataset is utilized as the testing set, and the remaining subset is used as the training set. The testing set receives a portion of the dataset for each of the four (4) attributes, based on the research's specifications and the size of the dataset. For this work, 70% of the dataset was used for training the model, and 30% was left aside for testing. For each of the six features, the dataset was divided into two, as seen in Figure 7.

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	Age
371	0	118	64	89	0.0	21
527	3	116	74	105	26.3	24
26	7	147	76	0	39.4	43
729	2	92	52	0	30.1	22
656	2	101	58	90	21.8	22
...	...	...	...	...	...	...
459	9	134	74	60	25.9	81
257	2	114	68	0	28.7	25
349	5	0	80	0	41.0	37
70	2	100	66	90	32.9	28
629	4	94	65	0	24.7	21

Figure 7: Data splitting for the six (6) Features

The dataset for the six features was also divided into training and testing sets using a technique known as "train-test split." A random subset of the dataset is divided into the training set when the train-test split is used, and the remaining subset is used as the testing set. Based on the work's requirements and the dataset's size, a percentage of the dataset for the six features is given to the testing set. 30% of the dataset for this work was also reserved for testing, whereas the remaining 70% was used for model training. Figure 7 shows the python commands for training the logistic Regression Model was trained.

```
In [11]: from sklearn.linear_model import LogisticRegression

         model = LogisticRegression()
         model.fit(x_train,y_train)
```

Figure 8: Training of Logistic Regression modelAfter splitting the data into Test and Train, the next step is to train the model. In Machine learning, when you train a model, it simply means you are adjusting the parameters of the model so that its performance in solving a certain task increases. The figure in 8 above shows how the model is being trained.

4.4 Tuning the Parameters

After training the model, the next step is to tune the parameters in order to optimize the performance. This is done to ensure that the model gives an accurate result. Hyperparameter tuning is an essential part of controlling the behavior of a machine learning model. If we don't correctly tune our hyperparameters, our estimated model parameters produce suboptimal results, as they don't minimize the loss function. This means our model makes more errors. In practice, key indicators like the accuracy or the confusion matrix will be worse.

In this implementation, this is done by selecting at row of different features in the data set table and placing it into the model in order to get the result which will be displayed in binary (1,0). In this case; 1 represents True and 0 represents False. Figure 9 shows how we tune the parameters to optimize the performance for the First Four Features

```
In [ ]:  x_test

In [ ]:  y_test

In [14]: i = model.predict_proba([[4,141,27.6,40]])
         print(i)

         [[0.61415724 0.38584276]]

In [15]: i = model.predict([[4,141,27.6,40]])
         print(i)

         [0]

In [16]: i = model.predict_proba([[13,106,34.2,52]])
         print(i)

         [[0.51079688 0.48920312]]
```

Figure 9: Tuning the Parameters of Four Features

The figure above shows the parameter tuning of the four features. It also shows the testing and probability detection of a row of features. The values for the first Four Features are (4, 141, 27.6, 40) and the result obtained was: 0 which means false and it has a probability of 0.61. Figure 10 shows how we tune the parameters to optimize the performance for Six Features.

```
         Name: Outcome, Length: 231, dtype: int64

In [38]: i = model.predict_proba([[4,141,27.6,40,34,12]])
         print(i)

         [[0.4473647 0.5526353]]

         C:\Users\SUNNAH\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but Logistic
         Regression was fitted with feature names
           warnings.warn(

In [39]: i = model.predict([[4,141,27.6,40,34,12]])
         print(i)

         [1]

         C:\Users\SUNNAH\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but Logistic
         Regression was fitted with feature names
           warnings.warn(

In [45]: i = model.predict_proba([[13,106,34.2,52,23,10]])
         print(i)

         [[0.81542325 0.18457675]]
```

Figure 10: Tuning the Parameters of six Features

The figure above shows the parameter tuning of the six features. It also shows the testing and probability detection of a row of features. The values for the six features are (4, 141, 27, 6, 40, 34, 12) and the result obtained was 1 which means true and it has a probability of 1.

Table 1: Performance Evaluation of the model on the selected metrics

	Precision	Recall	F1-Score	Support
0	1.00	0.88	0.93	8
1	0.92	1.00	0.96	12
Accuracy			0.95	20
Macro avg	0.96	0.94	0.95	20
Weighted avg	0.95	0.95	0.95	20

In Table 1, an evaluation of the performance of the binary classification model was presented. The model predicts two classes, labeled as '0' and '1'. For class '0', the precision is 1.00, indicating that when the model predicts class '0', it is correct every time. For class '1', the precision is 0.92, meaning that when the model predicts class '1', it is correct in approximately 92% of the cases. For class '0', the recall is 0.88, indicating that the model correctly identifies 88% of the actual '0' instances. For class '1', the recall is 1.00, meaning that the model correctly identifies all '1' instances. For class '0', the F1-score is 0.93, which is a good balance between precision and recall. For class '1', the F1-score is 0.96, indicating a strong balance between precision and recall. The 'support' column tells us how many instances are in each class. There are 8 instances of class '0' and 12 instances of class '1' in the test set. The overall accuracy of the model is 0.95, which is high. This means that the model correctly classifies 95% of the total instances in the test set.

4.4 Model Evaluation

The last stage in the experiment is evaluation of the model's performance. The metrics used to assess how well the model performs are accuracy, precision, sensitivity, F1 score and ROC Curve. The ROC curve is shown in Figure 11.

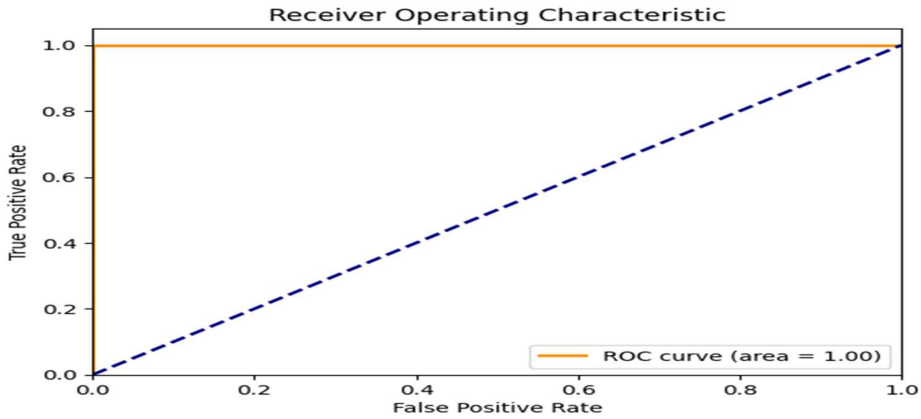

Figure 11: ROC Area of the model

5. RESULT AND DISCUSSION

Evaluating the model's performance is the last stage after the hyper-parameters have been adjusted and trained. Metrics like accuracy, precision, sensitivity, F1 score and AUC were used to assess how well the model performs. We measured the model's performance on a large dataset of patients with Type 2 diabetes in order to validate the model. With this implementation, accuracy metrics were used to assess the performance of the first four features. The results showed a 73% accuracy and a 27% cross-validation error rate, while accuracy metrics were also used to assess the performance of the second six features. These results showed a 79% accuracy and a 21% cross-validation error rate. A model's accuracy metrics for the Four Features were used to assess its performance, as shown in Figure 12.

```
In [ ]: x_test

In [ ]: y_test

In [14]: i = model.predict_proba([[4,141,27.6,40]])
         print(i)

         [[0.61415724 0.38584276]]

In [15]: i = model.predict([[4,141,27.6,40]])
         print(i)

         [0]

In [16]: i = model.predict_proba([[13,106,34.2,52]])
         print(i)

         [[0.51079688 0.48920312]]

In [17]: accuracy = model.score(x_test,y_test)
         print(accuracy)

         0.7359307359307359
```

Activate Windows

Figure 12: Evaluation of performance for Four Features

The evaluation of the four features of the data set gave us an accuracy of 73% and an error rate of 27%. The accuracy of the first four features is quite okay and it is enough to conclude that the model is efficient enough in the prediction of diabetes. Figure 13 shows how the performance of the model was evaluated with accuracy metrics for the Six Features.

```
In [46]: accuracy = model.score(x_test,y_test)
         print(accuracy)

         0.7922077922077922
```

Figure 13: Evaluation of performance for Six Features

The evaluation of the six features of the data set gave us an accuracy of 79% and an error rate of 21%. The accuracy of the six features is okay and it is enough to conclude that the model is efficient enough in the prediction of diabetes. After evaluating the performance of the model for the First Four Features and the Second Six features, we observed that there is an increase in the percentage of accuracy due to the increase in the number of features which means the higher the features, the higher the percentage of accuracy. This is summarized in Table 2 below.

Table 2: Comparing the effect of increased number of features

Number of features	Accuracy result	Cross validation error rate
4	73%	27%
6	79%	21%

From table 4.9.1, we are evaluating the performance of a binary classification model. The model predicts two classes, labeled as '0' and '1'. For class '0', the precision is 1.00, indicating that when the model predicts class '0', it is correct every time. For class '1', the precision is 0.92, meaning that when the model predicts class '1', it is correct in approximately 92% of the cases. For class '0', the recall is 0.88, indicating that the model correctly identifies 88% of the actual '0' instances. For class '1', the recall is 1.00, meaning that the model correctly identifies all '1' instances.

For class '0', the F1-score is 0.93, which is a good balance between precision and recall. For class '1', the F1-score is 0.96, indicating a strong balance between precision and recall. The 'support' column tells us how many instances are in each class. There are 8 instances of class '0' and 12 instances of class '1' in the test set. The overall accuracy of the model is 0.95, which is high. This means that the model correctly classifies 95% of the total instances in the test set.

However, the classification of the model based on the selected metrics shows that the model performs well for both classes, with high precision, recall, and F1-scores. The model is particularly strong at correctly identifying class '1' instances, achieving perfect precision and recall. The overall accuracy of 0.95 indicates the model's high performance in classifying the entire test dataset as against the result obtained in Jian et al., (2019), Poly *et al.,* (2023) and Israt et al., (2023).

6. CONCLUSION

One of the most important aspects of healthcare and illness prevention is identifying those who have a high risk of acquiring diabetes. This work presents a diabetes prediction model that provides deeper insights into the risk factors that can help with early diagnosis, high-risk individual classification, and the development of preventive and care methods. Our research has shown that the following factors: age, body mass index (BMI), family history, glucose levels, and frequency of pregnancy are highly predictive of Type 2 diabetes. The prediction accuracy of our suggested model is 73% and 79%, with matching cross-validation error rates of 27% and 21%.

With regard to classification trees, the tree with six nodes has been selected because it outperforms other possible sub trees in terms of prediction accuracy, coming in at 74.48%. Moreover, the model works well for both classes, with good precision, recall, and F1-scores, according to the performance evaluation of the classification based on the chosen metrics. The model achieves flawless precision and recall, and is especially good at properly detecting instances of class '1'. The model performs well in classifying the whole test dataset, as seen by its overall accuracy of 0.95. By taking the required steps to potentially lower the prevalence of Type 2 diabetes, this research highlights the potential impact of addressing these five prognostic indicators. Furthermore, the development of measures and the execution of healthcare policies targeted at preventing this disease may be influenced by the precise prediction of diabetes. With this thorough understanding of predictive factors, we can avoid disease and improve healthcare outcomes by being proactive.

REFERENCE

[1] Israt, J., Kakoly, M., Rakibul, H., & Najmul, H. (2023). Data-Driven Diabetes Risk Factor Prediction Using Machine Learning Algorithms with Feature Selection Technique. Sustainability, 15(6), 4930. https://doi.org/10.3390/su15064930

[2] Poly, T.N., Islam, M.M., Li, Y. J. (2022). Early Diabetes Prediction: A Comparative Study Using Machine Learning Techniques. *Studies in Health Technology and Informatics*. DOI: 10.3233/SHTI220752. PMID: 35773898.

[3] Wu, H., Yang, S., Huang, Z., Jian, H., & Wang, X. Y. (2020). Type 2 Diabetes Mellitus Prediction Model Based on Data Mining. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5983-5990). IEEE. https://doi.org/10.1016/j.imu.2017.12.006

[4] Tigga, N. P., & Garg, S. (2021). Predicting Type 2 Diabetes Using Logistic Regression. In Proceedings of International Conference on Artificial Intelligence: Smart Systems and Machine Learning (pp. 473-480). *Springer.* https://doi.org/10.1007/978-981-15-5546-6_42

[5] Dritsas, E. and Trigka, M. (2022). Data-Driven Machine-Learning Methods for Diabetes Risk Prediction. *Sensors (Basel),* 22(14), 5304. doi: 10.3390/s22145304. PMID: 35890983; PMCID: PMC9318204.

[6] Rajendra, P., & Latifi, S. (2021). Prediction of Diabetes Using Logistic Regression and Ensemble Techniques. *Computer Methods and Programs in Biomedicine*. https://doi.org/10.1016/j.cmpbup.2021.100032

[7] Yochai Edlitz and Eran Segal (2022) Prediction of type 2 diabetes mellitus onset using logistic regression-based scorecards *eLife* 11:e71862. https://doi.org/10.7554/eLife.71862

[8] Jian, L., Qin, H., Minghua, D., & Wei, Q. (2019). Construction of a risk prediction model of Type 2 Diabetes Mellitus based on logistic regression.

[9] Prianka Rajendra, Shahram Latifi. (2021). Prediction of diabetes using logistic regression and ensemble techniques. *Computer Methods and Programs in Biomedicine Update*. Vol. 1. 100032, ISSN 2666-9900, https://doi.org/10.1016/j.cmpbup.2021.100032.

[10] Neha Prerna Tigga, Shruti Garg. (2020). Prediction of Type 2 Diabetes using Machine Learning Classification Methods, *Procedia Computer Science,* Vol. 167, Pages 706-716. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2020.03.336.

[11] Nuankaew, P., Chaising, S., & Temdee, P. (2021). Average Weighted Objective Distance-Based Method for Type 2 Diabetes Prediction. *IEEE Access*, 9, 137015-137028. DOI: 10.1109/ACCESS.2021.3117269.

[12] Joshi, R. D., & Dhakal, C. (2021). Predicting Type 2 Diabetes Using Logistic Regression and Machine Learning Approaches. International Journal of Environmental Research and Public Health. https://doi.org/10.3390/ijerph18147346

[13] Babatunde R. S., Babatunde A. N, Balogun B. F., Umar E, Oke, A. A. & Obiwusi K. Y. (2023). A predictive system for Parkinson Disease using Generative Adversarial network (GAN). FUW Trends in Science & Technology Journal, ww.ftstjournal.com. e-ISSN: 24085162; p-ISSN: 20485170; Vol. 8 No. 3 pp. 381 – 390

[14] Babatunde R. S., Babatunde A. N, Balogun B. F. Yakubu I. A., Ogundokun R., Obiwusi K. Y & Umar E. (2023). A comparison of Boosting techniques for Classification of Microarray data. Ilorin Journal of Computer Science and Information Technology. Vol. 6, No. 2 pp.1-8(2023) ISSN: 2141-3959