A Novel Prediction Model for Diabetes Detection Using Gridsearch and A Voting Classifier between Lightgbm and KNN

Conference Paper · October 2021

DOI: 10.1109/GCAT52182.2021.9587551

5 authors, including:

Rashmi Rane
Dr. Vishwanath Karad MIT World Peace University
7 PUBLICATIONS **40** CITATIONS

SEE PROFILE

2021 2nd Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 1-3, 2021

A Novel Prediction Model for Diabetes Detection Using Gridsearch and A Voting Classifier between Lightgbm and KNN

Nachiket Dunbray
School of Computer Engineering and Technology,
MIT World Peace University,
Pune, India.
nick.dunbray@gmail.com

Rashmi Rane
School of Computer Engineering and Technology,
MIT World Peace University,
Pune, India.
rashmi.rane@mitpwu.edu.in

Sparsh Nimje
School of Computer Engineering and Technology,
MIT World Peace University,
Pune, India.
sparshn00@gmail.com

Jayesh Katade
School of Computer Engineering and Technology,
MIT World Peace University,
Pune, India.
jkatade@gmail.com

Shreyas Mavale
School of Computer Engineering and Technology,
MIT World Peace University,
Pune, India.
shreyasmavaleofficial@gmail.com

2021 2nd Global Conference for Advancement in Technology (GCAT) | 978-1-6654-1836-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GCAT52182.2021.9587551

***Abstract*—Diabetes is also known as diabetes mellitus is one of the world's most prominent health hazards of the current times. It is a chronic disease in which the pancreas isn't able to produce the right amounts of insulin for the body to absorb glucose into the body cells for energy and stays in the bloodstream in turn raising the blood glucose levels. If this is not detected and treated on time, it can affect other body organs as well and leads to organ failure thus becoming fatal.**

Machine learning and data mining are two emerging fields in today's tech world. With the help of these methods, we can observe the past data behaviors and can then predict the future outcomes to a certain extent. This brings rise to the term 'prediction models' that we all know in today's tech world. Keeping this in mind, we can create predictive models for diabetes prediction. This helps in the early detection of diabetes so that it can be treated at the earliest to avoid complications. By finding out the highest accuracy model, we can accurately predict whether the patient is diabetic beforehand and prevent further health issues. This research paper discusses the techniques that have been used to create a unique predictive model for the prediction of diabetes.

***Keywords*—Diabetes prediction, Machine learning, Diabetes mellitus, Data mining, Predictive model, Classification, Feature selection**

I. INTRODUCTION

A. Diabetes Prediction

In the present world, more people are tending to eat fast food and more fatty food by adding more oil to their cooking as well as eating more sugary foods. This has led to the depreciation in the population's health and wellbeing. More people are susceptible to various health issues like diabetes, heart attack, etc. This affects the body organs and eventually can be fatal causing devastating effects on family members and close friends. There are two commonly known types of diabetes – type 1 and type 2.

Type 1: The pancreas generates very little insulin. Doctors do not know what causes Type 1 diabetes. The immune system of the person may assault the pancreas, causing the insulin-producing beta cells to die. Genes may also have a role in certain people's circumstances.

Type 2: The body doesn't process the blood sugar (which is glucose) properly. Type 2 diabetes is caused by a mix of hereditary and lifestyle factors. Obesity or being overweight puts you at a higher risk. When you gain weight, your cells become more resistant to the effects of insulin on your blood sugar. This is a genetic condition. Genes in family members enhance their chances of acquiring Type 2 diabetes and being overweight. [1].

In 2019, IDF took a survey in which approximately 463 million adults between the ages of 20 – 80 years were diabetic [2]. More than 4.2 million people have lost their lives in the past year. More than 2.2 million children and adolescents have been detected with diabetes in the past year. 1 in 5 people who are above 65 years old has been detected with diabetes. 1 in 2 people who had diabetes was undiagnosed and unaware. Over 760 billion dollars have been spent on health expenses due to diabetes.

Taking into consideration of diabetes disease, we can apply Predictive Modeling (PM) to the diabetes data to predict whether a person is going to get diabetes or not based on certain categories. By predicting whether the person is going to be affected by diabetes, we can take preventive actions to help save the person's life. This will help contribute to saving people's lives in the long run.

B. Predictive Modeling Architecture

There are two broad steps in predictive modeling as shown in fig. 1:

1) Training phase

2) Testing phase [3].

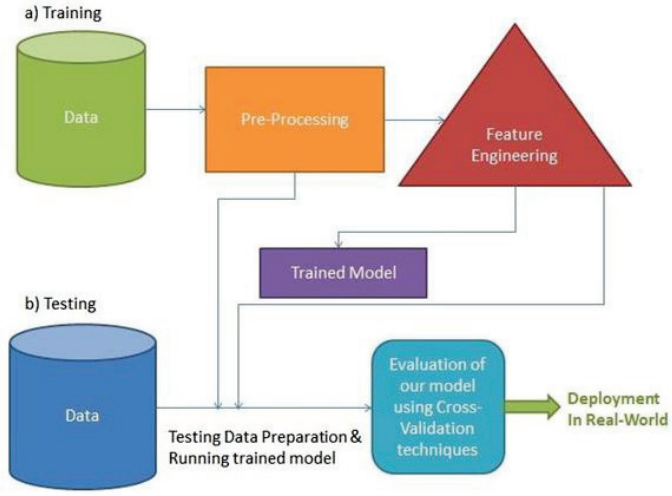

Fig. 1. Architecture of Predictive Modeling

1) Training phase

The training phase of predictive modeling is where the data is obtained, pre-processed, feature engineered and a training model is applied.

a) Obtaining of data: The data is obtained from various sources related to the problem statement. This data is collected through surveys, reports, forms, the internet, and many other resources. This data is raw and is unprocessed. We cannot infer much from this data as of now.

b) Pre-processing: The data that we obtain is in its raw form. There may be issues while collecting the data like some data might be missing, the data entered may not be true, etc. and hence this data is not useable. To obtain information from this data, we need to make it useable. Hence pre-processing is performed. Various techniques come under pre-processing. They are mainly categorized into three main steps; Data Cleaning, Data Transformation, and Data Reduction.

c) Feature Engineering: After we finish pre-processing the data, the data is almost useable. We need to convert the data into the proper form so that the various machine learning algorithms can accept the data. Each machine learning technique has its own set of requirements that need to be fulfilled for the algorithm to work. The feature engineering step also helps improve the performance of the machine learning algorithm as it suitable for usage.

d) Training the model: Once our data is prepared, we can finally apply machine learning models to train the data. We divide the total data into two main categories – training and testing. The model is trained using the training data. In this, the algorithm learns from past outcomes. It runs through the data and correlates the inputs to the outputs.

2) Testing phase

The testing phase involves the provision of test data to the trained model and performance analysis of the model. The data whose outcome we want to predict is given as input to the trained model. This is the testing data. It then gives the most accurate outcome according to its training. To understand the accuracy of the outcome, we evaluate the performance of the system.

C. Machine Learning Algorithms

Many machine learning algorithms are widely used for predictive modeling. Some of them are:

i. Linear Regression: This is primarily used for computing real values from continuous variables. The connection between independent and dependent variables is established by arranging them on a line. The regression line is the best-fitting line. We use this to calculate the cost of houses, total sales of a company, etc.

ii. Decision Tree: This is a well-known algorithm that is frequently used. It is mainly used as a classification method and works as a supervised learning algorithm. It can be used on both categorical and continuous variables for prediction.

iii. KNN: KNN, known as K – Nearest Neighbors, is used for regression as well as classification. It is a simple algorithm that classifies the data into the cluster and calculates the output based on the mean of the k clusters that are closest to that cluster [4].

II. LITERATURE REVIEW

K. VijiyaKumar and B. Lavanjya in 2019, worked on the dataset obtained from UCI Data Repository. They used Random Forest as their classifier to classify the data. Random forest is a supervised classification technique. It creates decision trees on data samples. It's an ensemble approach, which means it makes use of multiple decision trees instead of one decision tree to make its predictions. They performed a simple random forest. They did not mention the obtained accuracy but went on to mention that they were able to predict the data to some extent only, after which the model would fail [5].

In 2018, D. Dutta, D. Paul, and P.Ghosh analyzed important features that affect the prediction models. They used Logistic Regression, Support Vector Machine, and Random Forest techniques to test out their conclusions. They did their analysis of the Pima Indians Diabetes dataset. After analysis, they concluded by saying that glucose was the highest affecting factor which was followed by insulin. Both these features had extremely high importance in determining whether the person was diabetic or not. Then following these two came: age, BMI, and genes. These were not as significant as glucose and insulin but still was a determining factor for the predictions [6].

R. Syed, R. Gupta, and N. Pathik in 2018 did their research on an advanced tree adaptive data classification for diabetes disease prediction. They performed a multiclass classification

algorithm to increase the accuracy of the prediction. They did their work with Support Vector Machine and Tree-based algorithm with proper filtering. They applied SMORT to pre-process the data. They then applied their TPASVM to the data. Upon analysis, they found the accuracy to be 89.67 %. They had tested Random Tree, Random Forest, and J48 algorithms to compare their accuracies. They concluded that their TPASVM model gave the highest accuracy amongst them all. They said that their model was used for predicting Type – 2 diabetes [7].

P. Prabhu and S. Selvabharathi did their research to predict diabetes using the Deep Belief Neural Network in 2019. Deep Belief Neural Network (DBN) uses pretraining and fine-tuning to classify the data. They performed DBN on the Pima Indians Diabetes dataset from the Kaggle data repository. They said that by keeping minimal layers, the results were poor, and too many layers led to overfitting. Thus they found the optimal number of layers that led to an accurate prediction. They deduced that 4 layers were optimum. They compared their model to Naïve Bayes, Decision Tree, Logistic regression, Random Forest, and Support Vector Machine kernels. They obtained the highest accuracy of 80.8% which was of the DBN model. They concluded by stating that if their technique was further optimized, it would give a better accuracy [8].

In 2020, S. Patikar, P. Saha, S. Neogy, and C. Chowdhury proposed their research on Fuzzy KNN. They performed their research on the Pima Indian Diabetes dataset. They chose fuzzy KNN over KNN because the fuzzy membership value is determined by the Fuzzy KNN classifier. This membership value indicates how strongly a test sample is linked to a particular class. In the traditional KNN, we use the distances of the data points methodology. The fuzzy method is effective as it contains all the information about data. They used the Gaussian function to determine the membership function. On completing their implementation, they compared their algorithm to other algorithms and found out that theirs was the highest with an accuracy of 76.6% as compared to the normal KNN which had a mere accuracy of 71%. They tried the same implementation for another dataset that had only data about Type 1 diabetes patients and the accuracy came out to be 88.7% [9].

L. Loku, B. Fetaji, and M. Fetaji, in 2020, decided on creating their model using Artificial Neural Networks. Artificial Neural Networks mimic the biological neurons and have their perceptron. They used feed-Forward(), back-Propagate(), and The sigmoid(Val) methods to operate their neural networks. Their agenda was to see whether Neural Networks can be applied in the health care department to help the doctors and patients. They gave a comparative study of the attributes in their dataset that has the highest priority in detecting whether a person is diabetic or not [10]. M. Diab, S. Hussain, and A. Jarndal also used ANN in 2020 to predict diabetes. They used the FeedForward model and got an accuracy of 87.5%. They then used the Pattern model which had an accuracy of 86.7% [11].

N. Mohan and V. Jain primarily focused on Support Vector Machine in their research in 2020. They used Kaggle's Pima Indian Diabetes dataset for their research. They used the Linear, Polynomial, Radial Basis Function (RBF), and Sigmoid kernels of SVM. These kernels are mathematical functions that are predefined and are used for classification. Upon analysis, they found that the Radial Basis Function had the highest accuracy of 82% followed by the Polynomial kernel with 80% accuracy. The remaining models had a lower accuracy. They concluded that the RBF kernel can be used for many healthcare predictions as well [12].

R. Akula, N. Nguyen, and I. Garibay proposed their research work in 2019. They performed their research on two different datasets – Pima Indian Diabetes Dataset as well as Practice Fusion Dataset. They performed analysis on seven various algorithms and received interesting outcomes from them. According to their research, Decision Tree gave them the highest accuracy of 72% for the Pima Diabetes Dataset but for the Practice Fusion Dataset, the Neural Network model gave them the highest accuracy of 82.5%. They even tried boosting and ensemble models. On applying ensemble, they received higher accuracies. The Practice Fusion Dataset had an accuracy of 86% and the Pima Indians Diabetes dataset had an accuracy of 89.1% [13].

In 2020, A. Alanazi and M. Mezher surveyed the common machine learning algorithms which were used for the prediction of Diabetes. They focused on SVM and RF algorithms. According to them, the Random Forest algorithm got an accuracy of 87%. They performed their analysis on a Saudi Arabia healthcare institute [14].

S. Bhosale, S. Dagale, R. Lomte, and S. Ghodake surveyed many prediction models for diabetes detection in 2018. They compared Naïve Bayes, Support Vector Machine, C4.5, K – Nearest Neighbor, K – Means, Branch and Bound, Randomized Hill Climb, and Simulated Annealing. In their survey, they found that the Branch and Bound algorithm gives the highest accuracy along with C4.5 [15].

In 2020, M. K. Hasan, M. A. Alam, D. Das, E. Hossain and M. Hasan performed hyperparameter tuning on different algorithms and then made an ensemble model for each base algorithm. They even attempted to create a multilayer perceptron using the feedforward neural network technique. In the end, they found that their proposed ensemble model of (AB+XB) gave the highest accuracy[16].

N. L. Fitriyani, M. Syafrudin, G. Alfian and J. Rhee proposed an ensemble model as well. They used Multilayer Perceptron, Support Vector Machine, and Decision Tree as the initial classification level. For the second iteration they used Linear Regression. Their proposed model had an accuracy. They compared this to the other models used for comparison and found this to be the highest [17].

A. Zaitcev, M. R. Eissa, Z. Hui, T. Good, J. Elliott and M. Benaissa proposed a Deep Neural Network method to track the HBA1C in the body to predict diabeties. They were able to get a high confidence from their proposed method which was the highest from the compared ones. They performed 10 fold cross validation to evaluate the performance of the model [18].

In 2019, Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng and D. N. Davis proposed a classification algorithm that works on an imbalanced dataset with missing values. Their algorithm

was based in NB, ADASYN and RF algorithms. They evaluated their performance using k fold cross validation with k = 5 and 10. Their proposed model was able to achieve an accuracy of 87.10% [19].

III. METHODOLOGY

A. Predictive Modeling Approach

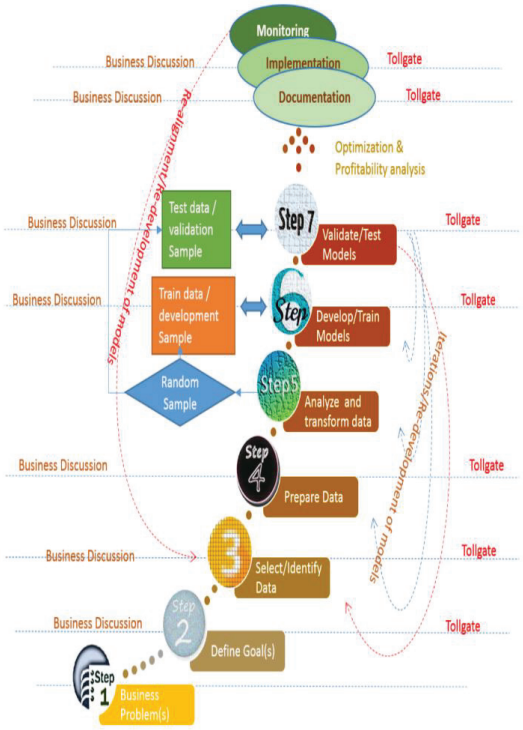

Fig. 2. Approach of Predictive Modeling

Fig. 2. Shows the predictive modeling approach or architecture that we follow to obtain our outcome. It is an iterative process and ends when we are satisfied with our outcome from the model

B. Machine Learning Technique

Machine learning is capturing the interest of researchers due to its accuracy and ability to adapt it according to the need. Predictive modeling mostly employs the supervised learning technique. It comprises the following stages: Data collection, Pre-processing, Training data, Testing data, and plotting results.

Data Collection: We have used the Pima Diabetes Dataset from the Kaggle data repository which is open to all.

Pre-processing: While observing the data graphically, we found that there are missing values. We encoded these missing values with null values (NaN). Thereafter, we filled them with the medians of the feature. For the prediction of diabetes, we initially had to find the correlation of the features with one another. Further, preparation of the data was done. For this Standard Scaling and Label Encoding was done. In Standard Scaling, we calculate the mean and scale the features to unit variance. The Label Encoder is used to encode labels with a value between 0 and n_classes-1. Finally, the correlation matrix was plotted.

Training data: To train the data, we used LightGBM (also known as Light Gradient Boosting Machine) which is a gradient boosting framework. Boosting is done to convert weak learners into strong learners. A weak learner is usually an algorithm whose accuracy is lesser than 50 %. It is an iterative method of generating new weak prediction rules using a base learning algorithm. After several rounds, the boosting algorithm merges these weak rules into a single powerful prediction rule.

Gradient boosting sequentially trains the models. Using the Gradient Descent technique, we use the new boosted models to gradually reduce the loss function ($y = a * x + b + e$) of the entire system. The repeated procedure yields a more accurate response variable estimate. The goal is to create new base learners that are maximally correlated with the loss function's negative gradient and are associated with the whole ensemble.

We combined KNN along with LightGBM using a voting classifier to increase the accuracy. K Nearest Neighbors implements datapoints and the Euclidean distance between the source point and the plotted points to figure out the K closest values, K being the integer input given to the classifier. VotingClassifier is a meta-classifier that allows you to combine similar or conceptually distinct machine learning classifiers for majority or plurality voting categorization. To improve the accuracy of the Voting Classifier, we use GridSearch CV to find the best "n neighbors." So the final algorithm is a combination of GridSearch + LightGBM and KNN.

Testing data: Cross-validation with k folds is done for the testing of data. Cross-validation is used to mainly understand the accuracy of the model based on data that hasn't been seen before. We utilise a small sample size to evaluate how well the model will perform in general when making predictions on unknown data.

For our application, we took k = 7. For KNN, we took nearest neighbors as 7 as it was obtaining a higher accuracy.

C. Algorithms

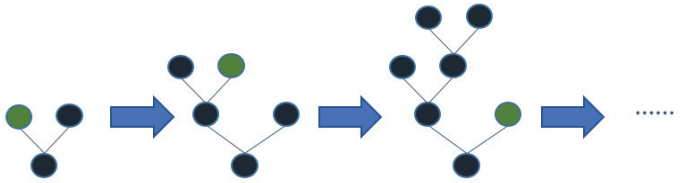

Fig. 3. Tree Growth in LightGBM algorithm

LightGBM is a tree-growth algorithm that works on a leaf-by-leaf basis. Other boosting algorithms build trees level-wise, but the LightGBM tree develops leaf-wise. The leaf with the greatest delta loss is selected for growth. Fig. 3. Depicts the growth of the leaf-wise tree.

The LightGBM algorithm
Input:
Training data: D = {(χ1, y1), (χ2, y2), ..., (χN, yN)}, χi ∈χ, χ ⊆ R, yi∈{−1,+1}; loss function: L(y, θ (χ));
Iterations:
M; Big gradient data sampling ratio: a; slight gradient data sampling ratio: b;
1: Combine features that are mutually exclusive (i.e., features never simultaneously accept nonzero values) of χi, i = {1, ...,N} by the exclusive feature bundling (EFB) technique;
2: Set $\theta_0(\chi) = arg\ min_c \sum_i^N L(y_i, c)$;
3: For m = 1 to M do
4: Calculate gradient absolute values:

$$r_i = \left| \frac{\partial L(y_i, \theta(x_i))}{\partial \theta(x_i)} \right|_{\theta(x)=\theta_{m-1}(x)}, i = \{1, \dots, N\}$$

5: Resample data set using gradient-based one-side sampling (GOSS) process:
$topN = a \times len(D)$; $randN = b \times len(D)$;
$sorted = GetSortedIndices(abs(r))$;
$A = sorted[\ 1 : topN]$; $B = RandomPick(sorted[\ topN : len(D)]\ , randN)$;
$\acute{D} = A + B$;
6: Calculate information gains:

$$V_j(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} r_i + \frac{1-a}{b}\sum_{x_i \in B_l} r_i\right)^2}{n_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} r_i + \frac{1-a}{b}\sum_{x_i \in B_r} r_i\right)^2}{n_r^j(d)} \right)$$

7: Develop a new decision tree $\theta_m(x)'$ on set D'
8: Update $\theta_m(\chi) = \theta_{m-1}(\chi) + \theta_m(\chi)$
9: End for
10: Return $\tilde{\theta}(x) = \theta_M(x)$

Fig. 4. LightGBM algorithm

Fig.4. above is the main algorithm that is followed during the LightGBM method. It is the pseudocode of the actual algorithm.

One of the most well-known and straightforward machine learning techniques is K Nearest Neighbour. It is a method of learning that is supervised. The KNN algorithm compares new data to old data and groups them together based on their similarities. This implies that as fresh data comes in, the KNN algorithm can quickly classify it into the appropriate categories. Consider the following scenario: We have a picture of an animal that resembles both a cat and a dog, but we don't sure what it is. So to identify it, we can use the KNN algorithm. The KNN model finds similar features to the cats and dogs images. It then categories it as either a cat or dog based on the similarity in features.

Fig. 5. It depicting the flow of the K Nearest Neighbors algorithm that is being used. It shows the exact flow that has been followed during the implementation of the algorithm.

IV. RESULTS

A. Dataset

This dataset is obtained from the National Institute of Diabetes and Digestive and Kidney Diseases. It consists of 768 patient records with the following details:

- Pregnancies: Number of times the patient was pregnant
- Glucose: A 2 hours concentration of plasma glucose of the patient

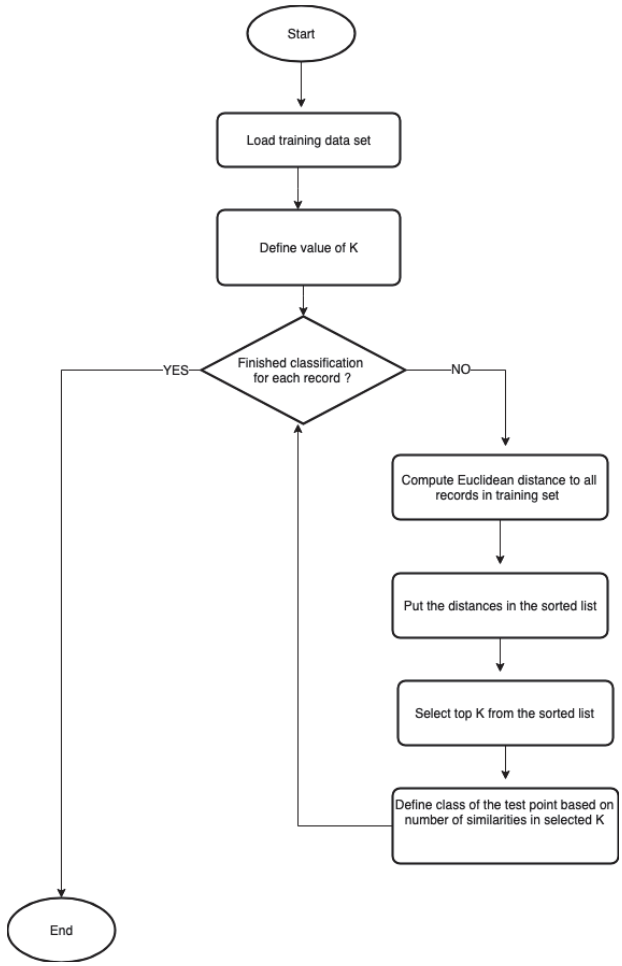

Fig. 5. Flow of KNN algorithm

- BloodPressure: Diastolic blood pressure of the patient (in mm Hg)
- SkinThickness: Triceps skinfold thickness of the patient (in mm)
- Insulin: 2-Hour serum insulin of the patient (in mu U/ml)
- BMI: Body mass index of the patient (in weight in kg/(height in m)^2)
- DiabetesPedigreeFunction: Diabetes pedigree function of the patient
- Age: Age of the patient (in years)
- Outcome: Class variable (0 or 1: Non-diabetic or diabetic)

B. Performance Analysis

Results were obtained and graphically represented. The confusion matrix was obtained along with the performance metrics. The ROC curve, as well as the precision-recall curve, was also displayed.

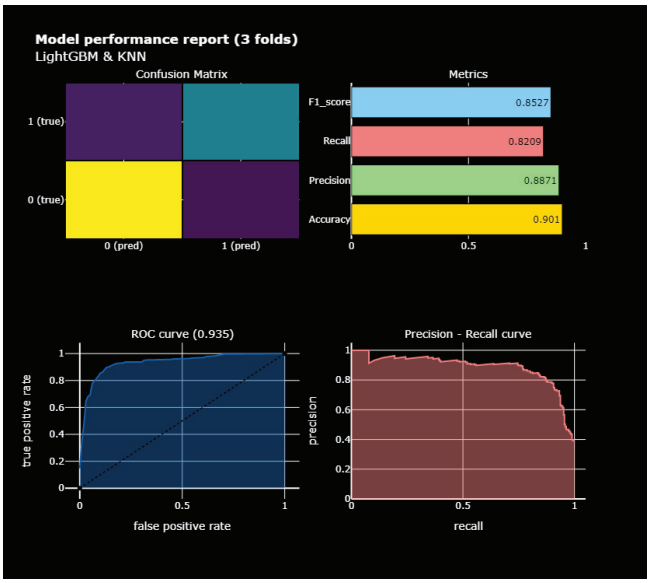

Fig. 6. Model Analysis Report

The success or failure of research is measured using several factors in a performance analysis. It aids in the development of a positive project management culture that produces outstanding results.

The initial implementation of simple KNN was able to predict upto an 86% accuracy. The standalone LightGBM model was able to predict upto 88.7%. But together, as observed, we have received an accuracy of 90.1%. Hence the voting classifier between the two has provided to be useful to increase the accuracy of the model.

Table I below, depicts a comparative analysis of some of the prior models that have been implemented and the proposed model. As we can see, it has obtained the highest accuracy measured from the set.

TABLE I. COMPARISON BETWEEN PREDICTION MODELS

MODEL	ACCURACY
KNN	86%
LightGBM	88.7%
SVM [8]	80.8%
Fuzzy KNN [9]	76.6%
Random Forest [14]	87%
DMP_MI [18]	87.5%
LightGBM + KNN	**90.1%**

Cross Validation - 3 folds
LightGBM & KNN

Fold	Accuracy	Precision	Recall	F1 score	Roc auc
1	0.89	0.878	0.796	0.835	0.931
2	0.915	0.917	0.83	0.871	0.958
3	0.928	0.938	0.849	0.891	0.95
mean	0.901	0.889	0.821	0.853	0.937
std	0.017	0.036	0.021	0.024	0.014

Fig. 7. : Analysis per Fold for Cross-Validation

The confusion matrix provides a matrix as the output. It gives the entire performance analysis of the model. It gives us the True Positives, True Negatives, False Positives, False Negatives. True positives are when the model predicted yes when the output was yes (475) whereas false positive is when the model predicted yes when the output was no (33). Similarly, the true negatives are when the model predicted no when the output was no (235) and false negatives are when the model predicted no when the output was actually yes (25). This is all displayed in a 2x2 matrix. The accuracy is then calculated by adding the true values and then dividing the sum by the total sample. All of these values are depicted in fig. 8.

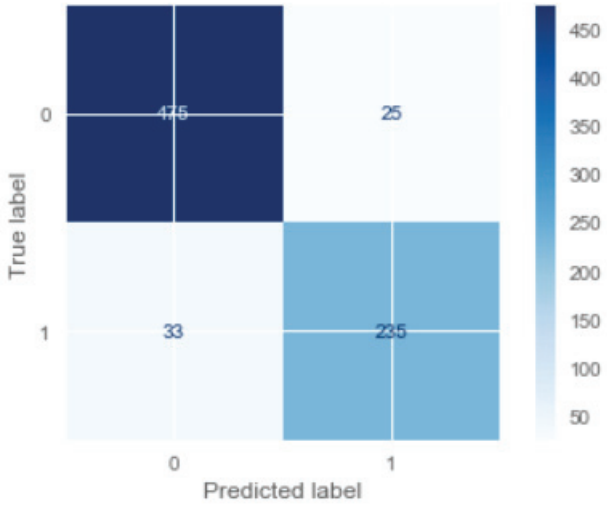

Fig. 8. Confusion Matrix with Values

V. CONCLUSION

Data processing and feature extraction methods were engaged in machine learning. Data analysis aids in prediction for a variety of reasons. Data mining algorithms have been developed in the past for effective categorization of illness and non-disease elements. The identification of a genetic abnormality at an early stage is one of the needed real-world medical concerns. Diabetes is a new disease that has become more prevalent among different generations of individuals. A better treatment can be facilitated by early diagnosis.

The Pima Indian diabetes database is taken into account and assessed. This research paper aims to develop and propose a model that will facilitate the prediction needs.In the research done, we were able to obtain an accuracy of 90.1%. This accuracy obtained is higher than the observed previous models that have been researched on and implemented by previous authors. This will help to improve the prediction accuracy for the patients. If we can further find more ways to enhance the models, we will have an enormous impact on the lives of those affected by diabetes. These same algorithms can be modified to early detect other different health issues.

REFERENCES

[1] G. G. Warsi, S. Saini and K. Khatri, "Ensemble Learning on Diabetes Data Set and Early Diabetes Prediction," 2019 International Conference on Computing, Power and Communication Technologies (GUCON), 2019, pp. 182-187.

[2] K. Kantawong, S. Tongphet, P. Bhrommalee, N. Rachata and S. Pravesjit, "The Methodology for Diabetes Complications Prediction Model," 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON), 2020, pp. 110-113

[3] A. M. Posonia, S. Vigneshwari and D. J. Rani, "Machine Learning based Diabetes Prediction using Decision Tree J48," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 498-502.

[4] V. S. Lakshmi, V. Nithya, K. Sripriya, C. Preethi and K. Logeshwari, "Prediction of Diabetes Patient Stage Using Ontology Based Machine Learning System," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2019, pp. 1-4

[5] K. VijiyaKumar, B. Lavanya, I. Nirmala and S. S. Caroline, "Random Forest Algorithm for the Prediction of Diabetes," 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 2019, pp. 1-5.

[6] D. Dutta, D. Paul and P. Ghosh, "Analysing Feature Importances for Diabetes Prediction using Machine Learning," 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2018, pp. 924-928.

[7] R. Syed, R. K. Gupta and N. Pathik, "An Advance Tree Adaptive Data Classification for the Diabetes Disease Prediction," 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), 2018, pp. 1793-1798.

[8] P. Prabhu and S. Selvabharathi, "Deep Belief Neural Network Model for Prediction of Diabetes Mellitus," 2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC), 2019, pp. 138-142.

[9] S. Patikar, P. Saha, S. Neogy and C. Chowdhury, "An Approach towards prediction of Diabetes using Modified Fuzzy K Nearest Neighbor," 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), 2020, pp. 73-76.

[10] L. Loku, B. Fetaji and M. Fetaji, "Prevention Of Diabetes By Devising A Prediction Analytics Model," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1-4.

[11] M. S. Diab, S. Husain and A. Jarndal, "On Diabetes Classification and Prediction using Artificial Neural Networks," 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), 2020, pp. 1-5.

[12] N. Mohan and V. Jain, "Performance Analysis of Support Vector Machine in Diabetes Prediction," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020, pp. 1-3.

[13] R. Akula, N. Nguyen and I. Garibay, "Supervised Machine Learning based Ensemble Model for Accurate Prediction of Type 2 Diabetes," 2019 SoutheastCon, 2019, pp. 1-8

[14] A. S. Alanazi and M. A. Mezher, "Using Machine Learning Algorithms For Prediction Of Diabetes Mellitus," 2020 International Conference on Computing and Information Technology (ICCIT-1441), 2020, pp. 1-3.

[15] R. Lomte, S. Dagale, S. Bhosale and S. Ghodake, "Survey of Different Feature Selection Algorithms for Diabetes Mellitus Prediction," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5.

[16] M. K. Hasan, M. A. Alam, D. Das, E. Hossain and M. Hasan, "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers," in IEEE Access, vol. 8, pp. 76516-76531, 2020.

[17] N. L. Fitriyani, M. Syafrudin, G. Alfian and J. Rhee, "Development of Disease Prediction Model Based on Ensemble Learning Approach for Diabetes and Hypertension," in IEEE Access, vol. 7, pp. 144777-144789, 2019.

[18] A. Zaitcev, M. R. Eissa, Z. Hui, T. Good, J. Elliott and M. Benaissa, "A Deep Neural Network Application for Improved Prediction of HbA1c\text{HbA}_{\text{1c}} in Type 1 Diabetes," in IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 10, pp. 2932-2941, Oct. 2020.

[19] Q. Wang, W. Cao, J. Guo, J. Ren, Y. Cheng and D. N. Davis, "DMP_MI: An Effective Diabetes Mellitus Classification Algorithm on Imbalanced Data With Missing Values," in IEEE Access, vol. 7, pp. 102232-102238, 2019.