DIABETES PREDICTION USING MACHINE LEARNING

- *Aravinda Raman J*

Abstract:

With the emerging increase of diabetes, which recently affected around 346 million people, more than one-third go undetected in the early stage, a strong need for supporting the medical decision-making process is generated. A number of researches have focused either on using one of the algorithms or on comparing the performances of algorithms on a given, usually predefined and static datasets that are accessible through the Internet. This paper focuses on implementing the k nearest neighbors (KNN), Random Forest Classifier, and Artificial Neural Network (ANN). The dataset contains some attributes that have not been previously used in computer-based evaluations.

Keywords- algorithms; diabetes; machine learning; KNN; Random Forest Classifier; ANN;

Introduction:

Diabetes is one of the deadliest diseases in the world. It is not only a disease but also the creator of different kinds of diseases like heart attack, blindness, etc. The normal identifying process is that patients need to visit a diagnostic center, consult their doctor, and sit tight for a day or more to get their reports. Diabetes is a disease that occurs when one's blood glucose, also called blood sugar, is too high. Blood glucose is the primary energy source and comes from our food. Insulin, a hormone made by the pancreas, helps glucose from food get into our cells to be used for energy. Sometimes our body does not make enough or any insulin or does not use insulin well. Glucose then stays in our blood and does not reach the cells. Over time, having too much glucose in our blood can cause health problems. Although diabetes has no cure as of now, one can take steps to manage diabetes and stay healthy.

The cause of diabetes varies depending on the genetic makeup, family history, ethnicity, health, etc. A blood test diagnoses diabetes & pre-diabetes.

Literature Survey:

In [1], Ayman et al. used the publicly available dataset from UCI and applied different machine learning classifiers to it. The classifiers they incorporated are Naive Bayes, Support Vector Machine, Random Forest, and Simple CART. Their approach starts with accessing the dataset, preprocessing it in the Weka tool, and then the 70:30 train and test split for applying different machine algorithms. The authors in [2] also used the publicly available dataset named Pima Indians Diabetes Database to perform their experiment. Their framework of performing the

prediction starts with the dataset selection and then with data preprocessing. Once the data was preprocessed, they applied three classification algorithms, i.e., naive Bayes, SVM, and Decision tree.

As they incorporated different evaluation metrics, they did compare the different performance measures and comparatively analyzed the accuracy. The highest accuracy achieved with their experiment was 76.30%. But the papers [1] and [2] do not have well defined preprocessing techniques, which inturn made their outcome not so accurate. The authors [3] proposed the neural network-based diabetes disease prediction on the Indians Pima Diabetes Dataset. They have used several hidden layers to find patterns in the data, and with the help of those patterns, they predicted the outcome. They name their proposed algorithms ADAP, a custom neural network with multiple partitions and a set of association weights and units. They managed to achieve a crossover point for sensitivity and specificity at 0.76 and are trying to precise their future results. The authors in [4] used diverse genera of machine learning algorithms like support vector machine, random forest, logistic regression, Decision tree, and many more on various types of disease datasets to show the applicability of Machine Learning in disease prediction and analysis. They also accompanied the traditional way of analyzing user data preprocessing, feature extraction and selection, classifiers training, and testing to produce the results. They used feature selection to reduce the computational expenses. Also, to get the most optimal outcome, they divided every dataset into a 90% training set and the remaining 10% testing set. This shows the importance of data preprocessing before implementing any machine learning algorithm. If the data is properly preprocessed, one can achieve higher accuracy.

Dataset Description:

This dataset is originally from the National Institute of Diabetes, Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether a patient has diabetes based on specific diagnostic measurements included in the dataset. Several constraints were placed on selecting these instances from a more extensive database. In particular, all patients here are females at least 21 years old of Pima Indian heritage

Dataset Link: https://www.kaggle.com/uciml/pima-indians-diabetes-database

There are 8 independent variables:

1. *Pregnancies:* No. of times pregnant
2. *Glucose:* Plasma Glucose Concentration a 2 hour in an oral glucose tolerance test (mg/dl)

Plasma Glucose Test	Normal	Prediabetes	Diabetes
2 hour post-prandial	Below 140 mg/dl	140 to 199 mg/dl	200 mg/dl or more

3. *Blood Pressure:* Diastolic Blood Pressure(mmHg)
    - If Diastolic B.P > 90 - means High B.P (High Probability of Diabetes)
    - Diastolic B.P < 60 - means low B.P (Less Probability of Diabetes)

4. *Skin Thickness:* Triceps Skin Fold Thickness (mm) –
   A value used to estimate body fat. Normal Triceps Skinfold Thickness in women is 23mm. Higher thickness leads to obesity and the chances of diabetes increase.

5. *Insulin:* 2-Hour Serum Insulin (mu U/ml)

	Normal Insulin Level
2 Hours After Glucose	16-166 mIU/L

   Values above this range can be alarming.

6. *BMI:* Body Mass Index (weight in kg/ height in ^{2}):

   Body Mass Index of **<18.5** is underweight.

   Body Mass Index of **18.5 to 24.9** is within the normal range.

   BMI between **25 and 29.9,** then it falls within the overweight range.
   BMI of **over 30 and < 34.9** falls within the obesity 1 range.
   BMI of **over 34.9 and < 39.9** falls within the obesity 2 range.
   BMI of **over 39.9** falls within the obesity 3 range.

7. *Diabetes Pedigree Function:* It provides information about diabetes history in relatives and the genetic relationship of those relatives with patients. Higher Pedigree Function means the patient is more likely to have diabetes.

8. *Age (years)*

9. *Outcome:* Class Variable (0 or 1) where '**0**' denotes the patient does not have diabetes, and '**1**' denotes the patient has diabetes

The **dependent variable (*Outcome*)** is whether the patient has diabetes or not.

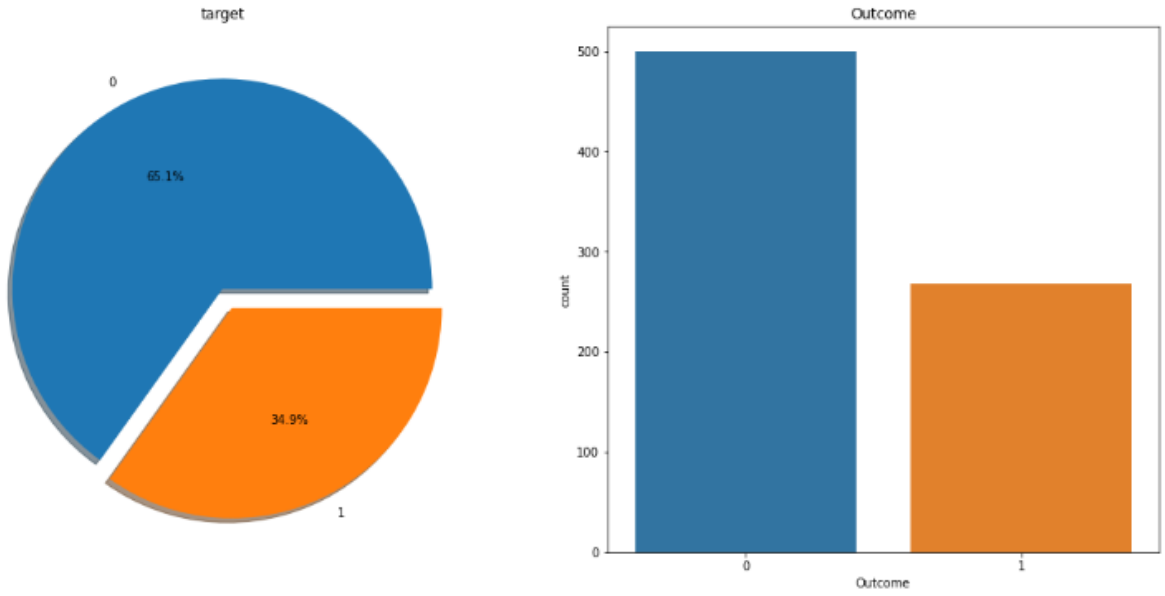

FIGURE 1. Number of people with and without diabetes

Sample Data

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0

Methodology:

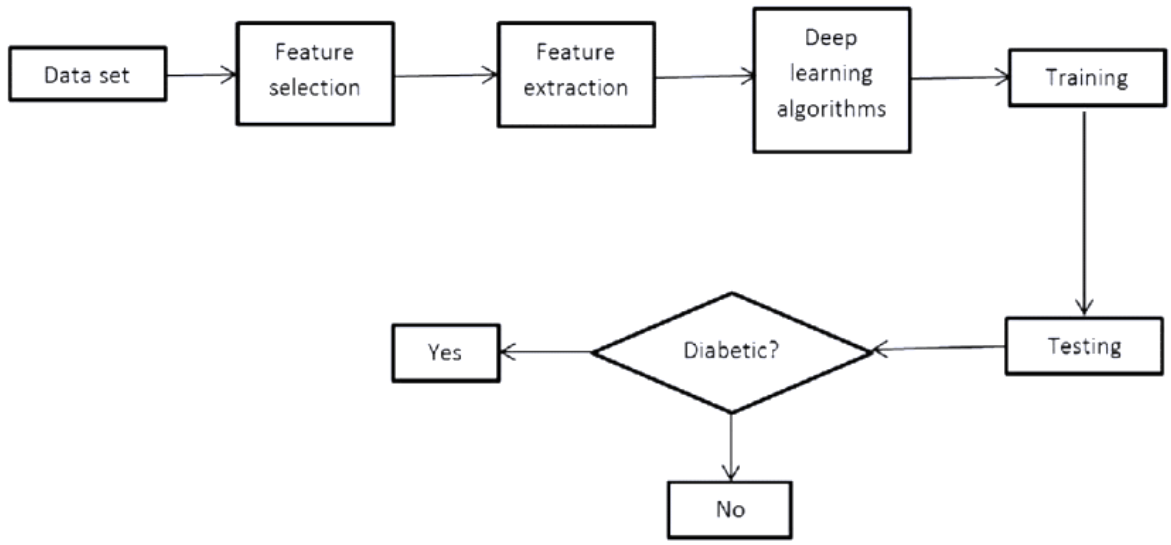

FIGURE 2. Flow diagram for diabetes prediction using a deep learning model

Data Cleaning will take place as the dataset has got a lot of missing values. It is very important to handle missing values because if not dealt with, they will reduce the overall prediction accuracy as a some of the data will be missing which helps in prediction. Handling missing values can be done by replacing null values with mode, mean, median or replacing the null value with a random variable. Here, the missing values in the dataset are replaced with the median of their respective attributes.

Data is then one-hot encoded using the limits of the attributes set in Dataset Description. One hot encoding is basically converting the categorical attributes to numerical attributes. One hot encoding makes our training data more useful and expressive, and it can be rescaled easily. By using numeric values, we can more easily determine a probability for our values. In particular, one hot encoding is used for our output values, since it provides more nuanced predictions than single labels.

RobustScaler is then used to scale the attributes so that the model does not get biased towards a particular attribute with a higher weight/data value. Robust scaler basically subtracts each data value by the median of that data value's attribute set and divides the result by the interquartile range (IQR).

The data is then split into training and testing sets. A training set and testing set ratio of 2/3 and 1/3 is used.

Metrics used to compare the models:

Precision: The precision can be defined as the number of TP upon the number of TP '+' number of FP. False positives are cases where the model is incorrectly tagged as positive that are actually negative.

Precision = TP / (TP + FP)

Recall: The recall can be defined as the number of true TP separated by the TP '+' FN.

Recall = TP / (TP + FN)

F1-Score: F1 is a function of Precision and Recall. F1 Score is needed when you seek a balance between Precision and Recall. There is an uneven class distribution (a more significant number of actual negatives).

F1= 2* (Precision*Recall) / (Precision + Recall)

K nearest neighbors (KNN):

KNN algorithm is a supervised machine learning algorithm that deals with similarity. KNN stands for K-Nearest Neighbors. It is a classification algorithm that will predict a class of a target variable based on a defined number of nearest neighbors. It will calculate the distance from the instance you want to classify to every instance of the training dataset. Then, it will classify the instance based on the majority classes of k nearest instances.

Choosing the number of neighbors is an essential step in the KNN algorithm, which ultimately determines the model's accuracy. So, we have plotted the value of accuracy vs. the number of neighbors (k) for a range of values to choose the best K value for our model.

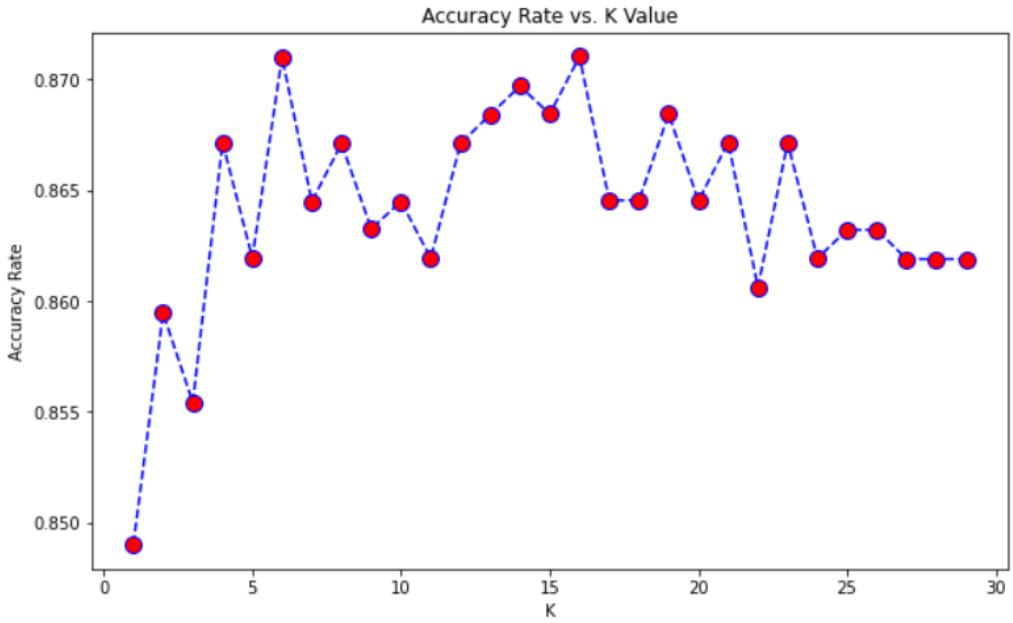

FIGURE 3. K vs. Accuracy rate

We are to select a value of K, after which there is not much decrease in accuracy, and primarily it only increases. Therefore, we pick the value of K as 23.

RESULTS:

An accuracy of 82.81% was obtained on the test data set.

A Precision of 87%, recall of 89%, and F1-score of 88% were obtained.

155 records were correctly classified as diabetes positive.
24 records were predicted falsely as diabetic.
20 records were falsely predicted as non-diabetic.
57 records were predicted correctly as non-diabetic.

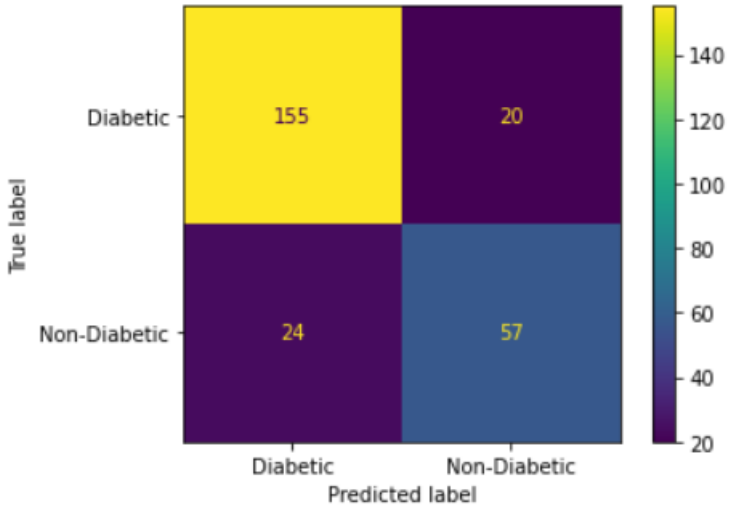

FIGURE 4. Confusion Matrix of KNN

Random Forest Classifier:

As its name implies, the random forest consists of an extensive range of individual call trees that operate as an associate degree ensemble. Every individual tree within the random forest spits out category prediction. Therefore, the class with the foremost votes becomes our model's prediction.

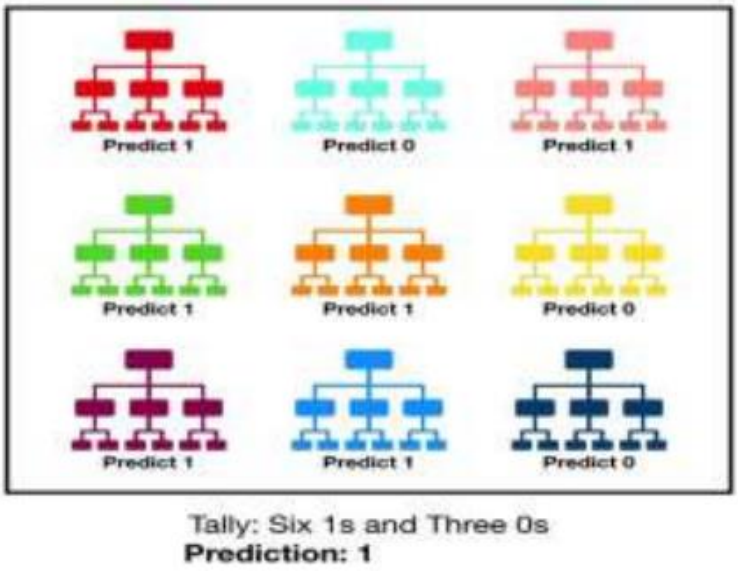

FIGURE 5. Random Forest Visualization

VISUALIZATION OF A RANDOM FOREST MODEL MAKING A PREDICTION

The fundamental concept behind random forest may be an easy, however powerful, one – the knowledge of crowds. In knowledge science-speak, the random forest model works well because an extensive range of comparatively unrelated models (trees) operational as a committee can beat out any individual constituent models. The low correlation between models is the key. Similar to investments with low correlations (like stocks and bonds) close to making a portfolio more significant than the total of its elements, unrelated models will turn out ensemble predictions that square measure additional correct than any of the individual predictions. The rationale for this glorious result is that the trees defend one another from their errors (as long as they do not perpetually all err within the same direction). Whereas some trees could also be wrong, several different trees will be correct; therefore, as a gaggle, the trees' square measure can move in the correct direction. That the conditions for a random forest to perform well are:

1. There has to be some actual signal in our options. Models engineered mistreatment those options do higher than random estimation.
2. The predictions (and so the errors) created by the individual trees got to have low correlations with one another.

RESULTS:

An accuracy of 87.89% was obtained on the test data set.

A Precision of 90%, recall of 93%, and F1-score of 91% were obtained.

163 records were correctly classified as diabetes positive.
19 records were predicted falsely as diabetic.
12 records were falsely predicted as non-diabetic.
62 records were predicted correctly as non-diabetic.

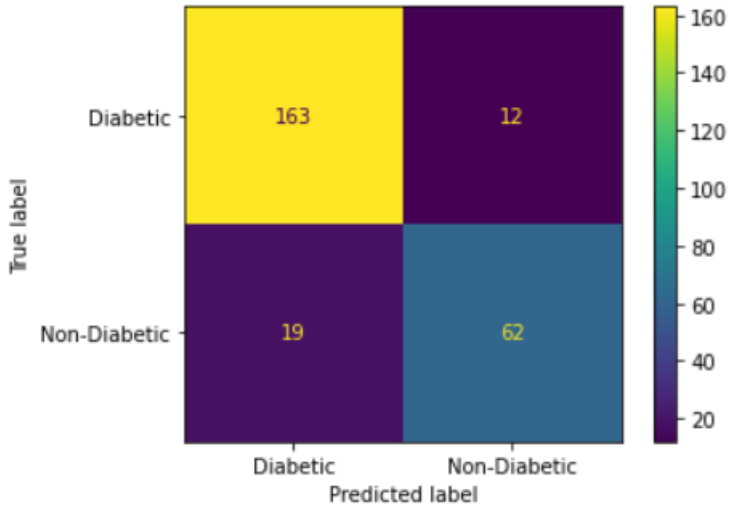

FIGURE 6. Confusion Matrix of Random Forest Classifier

Artificial Neural Network (ANN):

An ANN is a data processing system that consists of a large number of simple, highly interconnected processing elements in architecture inspired by the structure of the cerebral cortex portion of the brain. The neural network performs functions as the human nervous system. The brain processes information and thus tries to replicate the way humans learn. Neurons are the fundamental cellular unit of the brain. The neurons are responsible for receiving sensory information from the external world via dendrites, processing the information, and giving output via axons. Similarly, an artificial neural network consists of an input layer that consists of many neurons that takes the input, and an output layer that gives the output to the external world. In most cases, a hidden layer is present between the input and output layers which transforms the input into something which the output layer can use.

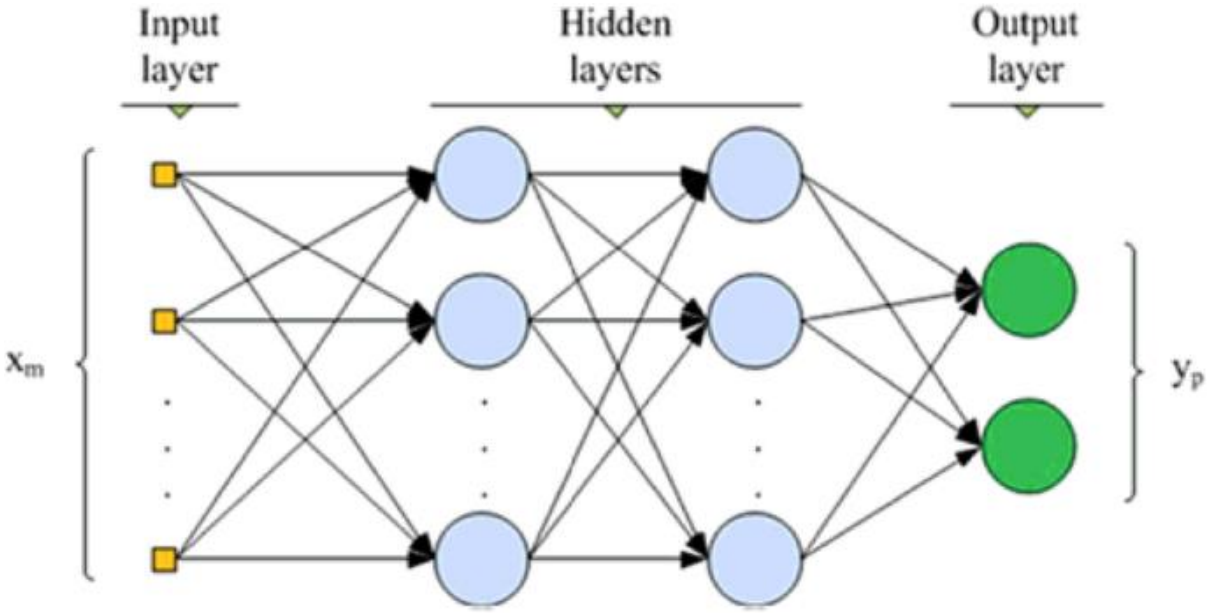

FIGURE 7. Neural Network

A. Neural Network Architecture

The neutral Network architecture consists of three layers: input, hidden, and output. The input layer takes simple inputs, the hidden layers where all the processing happens through a system of connections characterized by weights and biases. The output of our Neural network will either predict 1 or 0. In our architecture, we will only have 1 neuron in the final layer as it is a binary classification problem unlike Figure 7.

B. Backpropagation Algorithm

The Backpropagation algorithm takes the minimum value of the error function in weight space using the delta rule or gradient descent technique. The weights that minimize the error function are considered a solution to the learning problem.

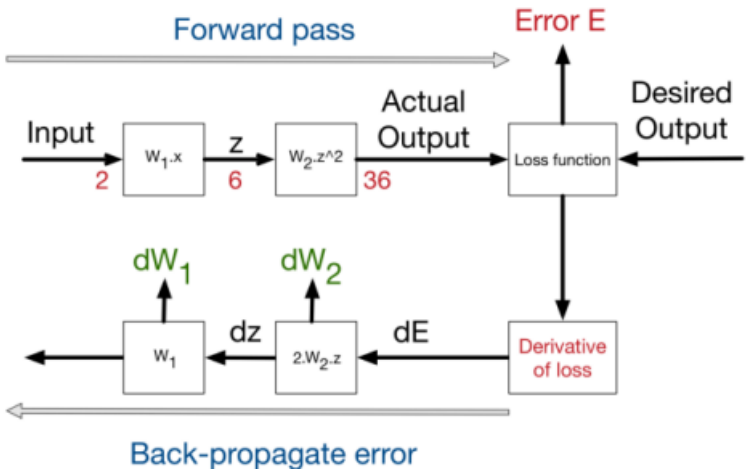

FIGURE 8. Working of Backpropagation Algorithm

In the neural network architecture, 7 hidden layers have been used. After every hidden layer, batch Normalization and Dropout of 30% have been used to prevent overfitting. The model has been trained for 100 epochs with Stochastic Gradient Descent as the optimizer and binary cross-entropy as the loss function. The final dense layer uses sigmoid activation function for classifying whether a person has diabetes.

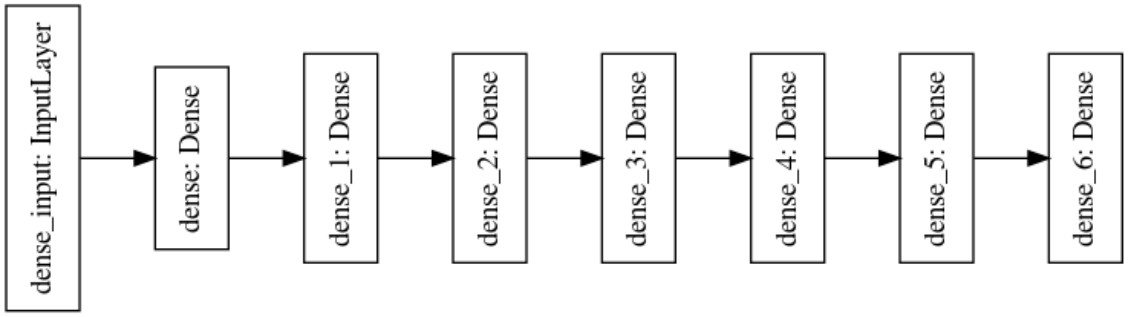

FIGURE 9. Neural network model architecture

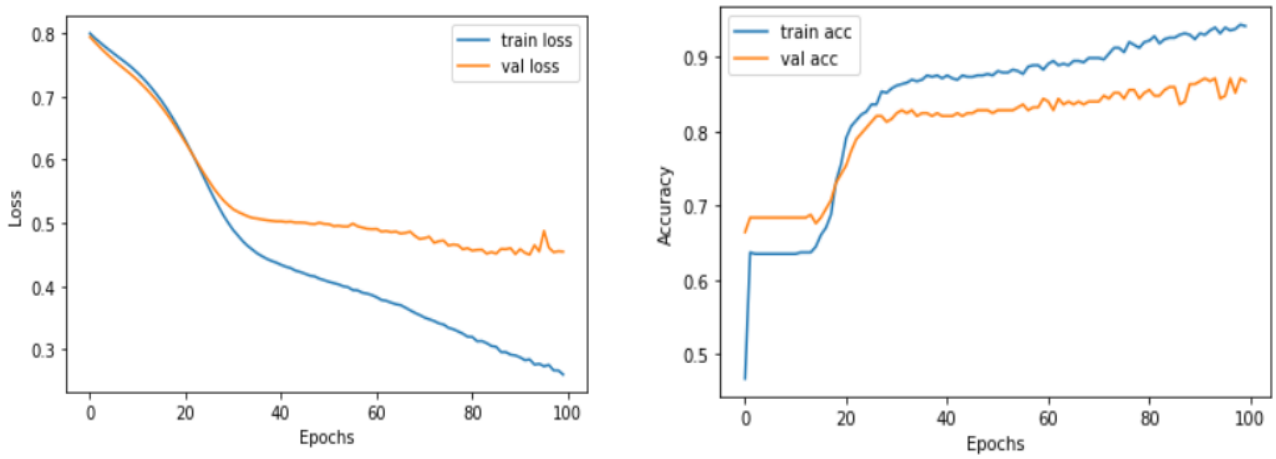

FIGURE 10. Epochs vs. (Loss and Accuracy)

RESULTS:

An accuracy of 86.32% was obtained on the test data set.

A Precision of 89%, recall of 91%, and F1-score of 90% were obtained.

160 records were correctly classified as diabetes positive.
20 records were predicted falsely as diabetic.
15 records were falsely predicted as non-diabetic.
61 records were predicted correctly as non-diabetic.

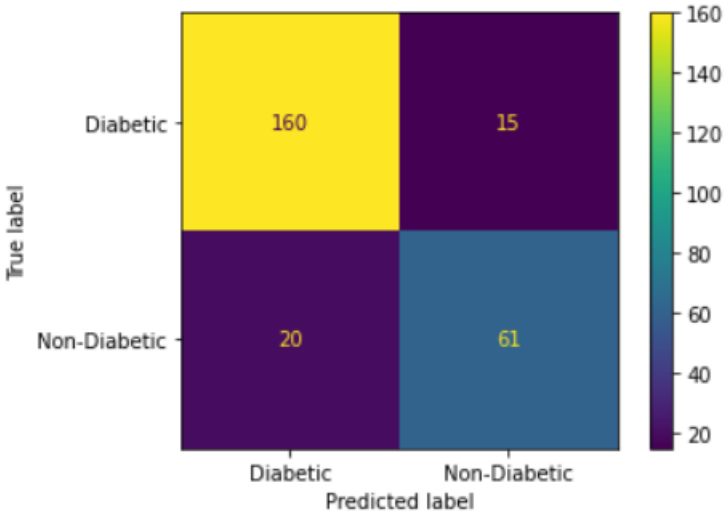

FIGURE 11. Confusion Matrix of Artificial Neural Network

Results:

MODELS	ACCURACY
K Nearest Neighbors	82.81%
Random Forest Classifier	87.89%
Artificial Neural Network	86.32%

Conclusion and future scope:

Random Forest Classifier gives us the highest accuracy of 87.89%, slightly higher than the artificial neural network model. The neural network can be further improved by training for more epochs, using more hidden layers, and a few more hyperparameter tunings. Ensemble models can also be used in the future to improve the results. Our proposed system eliminates the need to visit the clinic for diabetes diagnosis physically. As we have proposed and developed an approach for this using machine learning algorithms, it has significant potential in medical science for the detection of various medical data accurately. In the future, the Diabetes prediction algorithm can be made more efficient in terms of feature importance and new standardized dataset and correctly predicting the outcome since we are using Neural Networks and research to develop new algorithms which are more efficient than currently existing algorithms. Implementing these in our system will significantly enhance the performance of the system. Diabetes prediction can be improved further if more features are taken into consideration. Finally, time-efficiency can also be increased for various applications and integrations with other applications.

References:

[1] Mir, Ayman & Dhage, Sudhir. (2018). Diabetes Disease Prediction Using Machine Learning on Big Data of Healthcare. 1-6. 10.1109/ICCUBEA.2018.8697439.

[2] Deepti Sisodia, Dilip Singh Sisodia, Prediction of Diabetes using Classification Algorithms, Procedia Computer Science, Volume 132, 2018, Pages 1578-1585, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.05.122.
(https://www.sciencedirect.com/science/article/pii/S1877050918308548)

[3] J. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," Proceedings - Annual Symposium on Computer Applications in Medical Care, vol. 10, 11 1988.

[4] P. S. Kohli and S. Arora, "Application of machine learning in disease prediction," in 2018 4th International Conference on Computing Communication and Automation (ICCCA),

2018, pp. 1–4