International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 08 | Aug 2021 www.irjet.net p-ISSN: 2395-0072

Prediction of Diabetes using Machine Learning

Mayur Raj Singh Chouhan[1], Manoj Naik[1], Paurush Gupta[1], Nandini Mittal[1]

[1]Student ECE, RV College of Engineering, Bengaluru

Abstract - *Diabetes is one of the chronic diseases due to dysfunction in pancreas which causes the formation of less or no Insulin. The main cause of diabetes remains unknown, yet scientists believe that both genetic factors and environmental lifestyle play a major role. Patients suffering from diabetes develop many other diseases such as nerve damage and heart diseases. Hence, detection of this disease at an early stage can prevent complications and reduce several other health issues. In recent years, models have been built with an Area under ROC curve (AUC(ROC)) of 95% for the prediction of diabetes while the work proposed in the paper is to build an efficient machine learning model with an improved AUC(ROC) of 97.53% using feature creation which is a combination of existing features. The model is deployed over the cloud which improves accessibility with an easier user interface (UI). The model was tested and implemented on Jupyter Notebooks using Python with the usage of Pima Indian diabetes dataset given by the National Institute of Diabetes and Digestive and Kidney Diseases.*

Key Words: Diabetes Prediction, Data processing, Machine Learning, Web Deployment.

1.INTRODUCTION

Diabetes is the fastest growing lifestyle disorder in developing and developed countries [1]. Pancreas secrete insulin hormone which helps the body to absorb glucose from food and use it in body. The lack of that hormone causes diabetes which can result in Chronic diabetes conditions including type 1 diabetes and type 2 diabetes. Potentially reversible diabetes conditions include prediabetes and gestational diabetes. Prediabetes is a state when the sugar level in the body is higher than normal. And prediabetes is often the precursor of diabetes unless appropriate measures are taken to prevent progression. Gestational diabetes occurs during pregnancy but may resolve after the baby is delivered. [2]. Recent data on diabetes patients tells that diabetes among adults (over 18 years old) has risen from 4.7 % to 8.5 % in 1980 to 2014 respectively and is growing in other parts of the world too [3]. Data show that in 2017 more than 451 million people had diabetes worldwide, and will increase to 693 million in 20 years [4]. Another research [5] showed how widespread diabetes is, and reported that 500 million people have diabetes in the world, and this number is expected to reach 25 % and 51 % respectively in 2030 and 2045. Doctors confirm an early stage diagnosis helps in controlling diabetes to a great extent. But the identification process is cumbersome and involves diagnostic center visit and consultation with a doctor wastes time and the budget of health care systems and people every year.

In the past few years, many methods have been used and published for diabetes prediction. A Machine Learning (ML) based framework was proposed in [6] where authors implemented the Linear Discriminant Analysis [7], Gaussian Process Classification [8], Random Forest (RF) [9], Naive Bayes (NB) [10], Support Vector Machine (SVM) [11], Logistic Regression (LR) [12], AdaBoost (AB) [13], Quadratic Discriminant Analysis [14], Decision Tree [15], and Artificial Neural Network [16] with different dimensionality reduction techniques and cross-validation techniques. Researchers performed a few demonstrations where by rejecting outliers and padding values that are missing, they were able to get an Area Under Receiver Operating Characteristic (ROC) Curve AUC(ROC) 0f 0.930 in some of ML algorithms. In [18], paper we find 3 ML classifiers DT, NB and SVM to predict the probability of diabetes. They showed that NB performs best and has AUC of 0.819. Bagging ensemble and AB uses J48 (c4.5)-DT, as a learner(base) and for data mining (J48), have implemented [19] for the classification of diabetes.

In [20] genetic programming was used to get best parameters and outdo other techniques. Authors, in [21], employed bagging and boosting techniques to increase accuracy. Various algorithms give us a neat insight on prediction of diabetes using ML but further improvements can be done to increase its robustness. In this literature, use of pre-processing on Pima Indian dataset to reach our desired outcome by rejecting outliers, padding missing values, standardizing data. Replaced missing position of attribute by median value, due to a more central tendency for dataset. K folding is performed on a dataset to have the same class proportion, as the original PIMA dataset. implemented algorithms such as (k-nearest Neighbour (k-NN), RF, AB, and XGBoost (XB). Hyperparameters of ML models were improved using grid search. Maximized AUC to get the best result for our dataset. We use AUC since AUC is not biased in class distribution.

2. Proposed Methodology

The proposed method, illustrated in figure 1 and 2, includes data processing, model trained and deployment of model to cloud. The subsequent subsection gives the detailed information about the data processing, model training and cloud deployment.

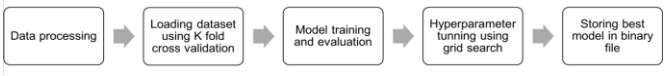


Fig -1: Block diagram of proposed methodology to build model

Fig -2: Block diagram for creating cloud-based web application

A) Data Processing:

Dataset: Data set was taken from UCI repository of machine learning database [11]. This dataset consists of 768 women suffering from diabetes among the Pima Indian population near Phoenix, Arizona. Training a neural network model requires a viable set of data. This dataset includes patients of both categories including 500 diabetes positive patients and 268 non diabetic patients along with various diagnostic attributes as mentioned in Table 1.

Table -1: Description of dataset attributes

Parameter	Description
Pregnancies	Number of times pregnant
Glucose	Glucose concentration in Plasma a in an oral glucose tolerance test
Blood Pressure	Blood pressure in mm Hg
SkinThickness	Thickness in Triceps skin fold (mm)
Insulin	Serum insulin in 2 hrs (mu U/ml)
BMI	Body mass index (weight in kg/(height in m)^2)
DiabetesPedigreeFunction	Diabetes pedigree function depending on genes
Age	In years
Outcome	Class variable (0 or 1) 268 of 768 are 1, the others are 0

Pre-processing of the data:

The absence of attributes in the dataset requires modification. Hence, the missing values such as blood pressure, skin thickness, BMI (body mass index) and age are replaced by calculating the median of the corresponding attributes as shown in the equation (1).

$$F(x) = \begin{cases} median(x) & x \text{ is null/zero} \\ x & otherwise \end{cases} \quad (1)$$

where x is a vector of attribute grouped based on outcome. The dataset also contains outlier which are deviated from the observation which need to be rejected. This is done using formula as in equation (2).

$$G(x) = \begin{cases} x & if\ Q_1 - 1.5 * IQR \le x \le Q_3 + 1.5 * IQR \\ null & otherwise \end{cases} \quad (2)$$

where x is the value of corresponding attribute and IQR is interquartile range and Q1 and Q3 are first and third quartile respectively of corresponding attribute. Now the null values are replaced with median using equation (1) given above.

Feature creation is performed by combining the pre-defined features due to the observations seen from the two dimensional scatter plot which consists of features on its axis as most of the results of a particular outcome falls under a common category which builds up to a new feature on its own to obtain better results comparatively. Existing eight features are used in permutations to form 16 new features using the feature creation methodology which gave us 24 features in total to test upon.

The data need to be rescaled so that mean and standard deviation of an attribute vector is zero and one respectively. This is achieved using equation (3).

$$(x) = \frac{x - \underline{x}}{\sigma} \quad (3)$$

where x is attribute vector, x` is mean of corresponding attribute vector and sigma is standard deviation.

B) Model training:

The dataset was partitioned using a K fold cross validation (KCV) technique. The performance metrics is calculated for each fold and the mean of all metrics is formulated as the final metric.

Different ML models were implemented on the processed dataset like logistic regression, random forest, k-NN, AdaBoost, XGBoost, Gaussian NB and the performance of each model was tested for all parameters including newly formulated features. All the models using the features were tested for performance metrics, each metrics having its own significance namely Sensitivity (Sn), Specificity (Sp), Precision (Pr), accuracy and AUC(ROC)[12]. Sn and SP are used for measuring type 1 and type 2 error [23]. The overall performance is calculated based on the best AUC results obtained from among the algorithms. After the algorithm is decided we run over an analysis to obtain the best set of features with a comparison performed based on AUC among the set of features of a particular algorithm. The selected model were then tuned for parameters using grid search. Grid search is implemented to improve AUC(ROC) performance metrics. Parameters are shown in table 2.

Table -2: Parameter of different ML model used for tuning using grid search

ML models	Hyperparameters
XGB Classifier	1. Gamma:Minimum loss reduction to make a partition on a leaf node.
	2. Max_Depth:Maximum depth of a tree
	3. Alpha:L1 regularization term on weights
	4. Lambda:L2 regularization term on weights
	5. subsample:Subsample ratio of the training instances.
	6. Colsample: This is a family of parameters for subsampling of columns.
Random Forest Classifier	1. Bootstrap:Sampling data points method
	2. Max_depth:Maximum levels in decision tree
	3. Max_features:Maximum features for splitting a node.

	4. Min_sample_leaf=Minimum data points in a leaf node.
	5. Minimum_sample_split=Minimum data points placed in a node before the node is split
	6. n_estimators=Number of trees
AdaBoost Classifier	1. Learning_rate:shrinks the contribution of each classifier
	2. n estimators:Maximum number of estimators when boosting is terminated.
K-Neighbors Classifier	1. leaf_size:Affect the speed of the constriction and query
	2. n_neighbors: neighbors to use by default
	3. weights: Function used in prediction

C) CLOUD FEASIBILITY AND ITS IMPLEMENTATION:

The model is deployed onto a cloud interface. The Front-end page was created using HTML and Flask which provides an UI. Further, this web application is integrated onto the cloud using Heroku.

Before the deployment of the application on Heroku the below two files are to be pushed along with the necessary project files onto Github for the further implementation process. The two files have been mentioned as follows-

1.Profile: Heroku apps include a Procfile that specifies the commands that are executed by the app on start-up.

2.Requirments.txt: Requirements.txt file is used for specifying what python packages are required to run the project.

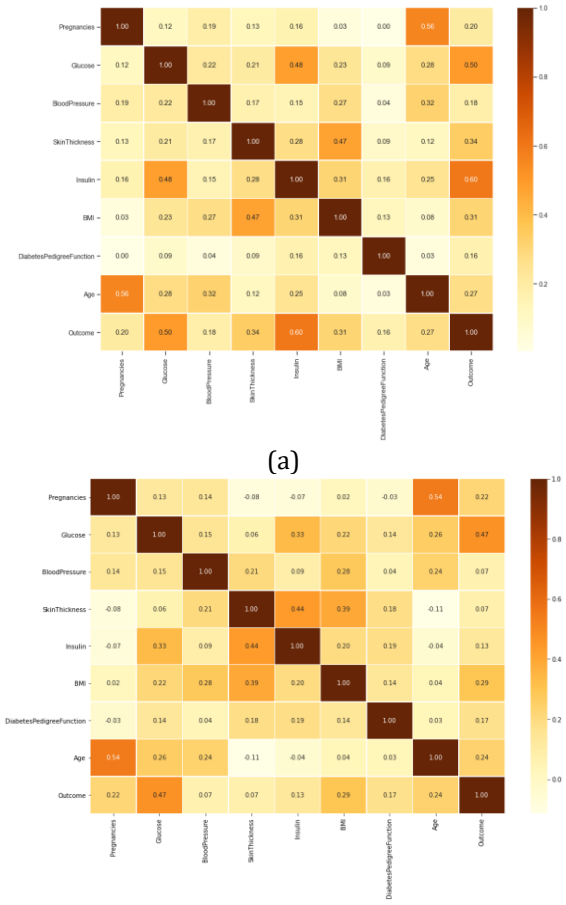
Integration of Github with Heroku is performed which is followed by deployment of our project onto the cloud platform, Heroku. A link is generated which provides global accessibility to the user to the experience the working of the framework.

3. Result and discussion

In this section the result obtained from different ML models are discussed. The subsection section describes the result of pre-processing, model training and cloud deployment respectively.

A) Result of pre-processing

After applying various data processing techniques, the dataset was improved in various aspects. The presence of Outliers resulted in an increased skewness of the data set and hence implementation of outlier rejection in the model helped eliminate this problem which resulted in low skewness. Feature creation helped improve the performance of the proposed model as it involved the combination of the potential features resulting in improvement of performance significantly. The standardisation of the dataset helped reduce the skewness even further. The correlation matrix of pre-defined features after data processing is shown in figure 3. The statistical difference between raw data and processed data is shown in Table 3.1 and Table 3.2 respectively.



(a)

(b)

Fig - 3: Correlation matrix of (a) raw data (b) processed data

In figure 3 it can be seen that with the help of data processing, correlation between attributes and outcome has improved, for the features such as insulin, skin thickness and blood pressure a significant improvement has been observed.

Table 3.1: Statistical description of raw data

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Table 3.2: Statistical description of processed data

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.00000	768.00000	768.00000	768.00000	768.00000	768.00000	768.00000	768.00000	768.00000
mean	-0.00000	0.00000	-0.00000	0.00000	0.00000	0.00000	0.00000	-0.00000	0.00000
std	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
min	-1.15827	-2.54979	-2.94882	-2.29593	-2.39054	-2.18032	-1.42711	-1.06174	-0.73164
25%	-0.85248	-0.71977	-0.74474	-0.28535	-0.47431	-0.73144	-0.75131	-0.79069	-0.73164
50%	-0.24089	-0.15353	-0.01004	-0.28535	-0.47431	-0.03037	-0.27326	-0.33893	-0.73164
75%	0.67649	0.60966	0.72465	0.55238	0.97640	0.63955	0.62883	0.65493	1.36501
max	2.81704	2.53816	2.92874	2.22786	3.15246	2.77392	3.11083	3.00405	1.36501

Table 3.1 and 3.2 show the statistical significance of data processing. To standardise data mean and standard deviation was set to zero and one respectively. With help of this skewness, the difference between mean and median is reduced.

B) Result of model training

There has been experimentation with different supervised classifiers. A variety of metrics are considered for evaluation. Finally, the best algorithm is compared with the state-of-the-art to validate our contributions in this literature.

Table 4 shows the best tuned parameter achieved with grid search for different algorithms. It can be observed from the data given in Table 4 that the performance of various models with best turned hyper parameters for different performance metrics.

Table -4: Different ML model performance (AUC(ROC)%) with hyper tuned parameters

ML model	Hyperparameters		Performance
XGB Classifier	1.	Gamma=0.9	97.539
	2.	Max_Depth:7	
	3.	Alpha:1.2	
	4.	Lambda=0.8	
	5.	subsample=0.8	
	6.	Colsample=0.3	
Random Forest Classifier	1.	Bootstrap=True	96.34
	2.	Max_depth=11	
	3.	Max_features=auto	
	4.	Min_sample_leaf=3	
	5.	Minimum_sample_split=5	
	6.	n_estimators=100	
AdaBoost Classifier	1.	Learning_rate=0.1	97.204
	2.	n estimators=450	
K Neighbors Classifier	1.	leaf_size=3	92.367
	2.	n_neighbors=17	
	3.	weights=distance	

From the data given in Table 5 and 6 it is observed that the proposed XB produces the best prediction for balanced metric of AUC(ROC), individually it is the best model for AUC(ROC), F1 score, and second best in Sp, Sn and Precision. Random Forest Classifier also works very close compared to XGB, it is better by a small margin in three metrics, Sn, Sp and accuracy.

Table 5: Precision F1 score Accuracy for different algorithm

Classifier	precision	F1-Score	Accuracy
XGB classifier	0.675	0.787	91.020
Random Forest Classifier	0.671	0.778	90.367
AdaBoost Classifier	0.671	0.787	92.061
K Neighbors Classifier	0.6789	0.780	88.023

Table -6: Specificity, Sensitivity and AUC(ROC) for different algorithm

Classifier	Specificity (Sp)	Sensitivity (Sn)	AUC(ROC)%
XGB classifier	0.847	0.944	97.539
Random Forest Classifier	0.843	0.926	96.34
AdaBoost Classifier	0.865	0.950	97.204
K Neighbors Classifier	0.809	0.918	92.367

The proposed classifier XB is performing best for the dataset used and gives a result of 97.539 % AUC(ROC). The best model (XB), along with our proposed pre-processing (missing value filling and standardisation), is tremendously successful for prediction of diabetes on the dataset. The data given in Figure 4 shows the confusion matrix of optimal XB model.

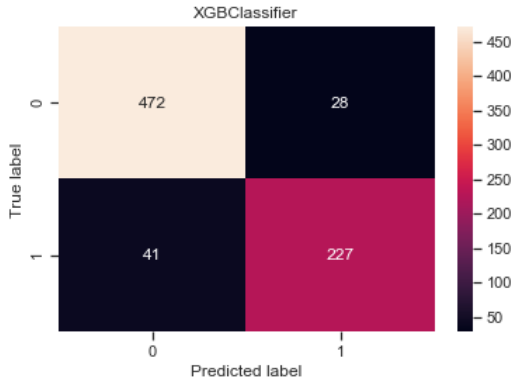


Fig- 4: Confusion matrix of XGBoost tuned model

C) Cloud deployment

The best trained model was used to predict the probability of diabetes. The model was deployed on Heroku with help of flask, HTML and CSS.

On opening the web application, a form can be seen, as shown in Figure 5, to fill eight diagnostic attributes.

Fig- 5: UI of cloud-based web application

Fig- 6: Result page of web application

While submitting the form, page is redirected to the output page showing the probability of having Diabetes according to the data entered as shown in Figure 6.

Performance Comparison: The performance of the work proposed in the paper is compared with the papers given in the literature and is tabulated shown in Table 7.

Table -7: Performance Comparison

Author and year	Specificity	Sensitivity	AUC(ROC)%
D. Sisodia et al. (2018) [17]	-	763	81.9
M. Maniruzzaman et al. (2018) [6]	0.797	0.96	93.0
Q. Wang et al. (2019) [24]	-	0.854	92.8
S. P. Chatrati et al. 2020 [25]	0.76	.72	70.0
MD. KAMRUL HASAN, MD. ASHRAFUL ALAM 2020 [23]	0.934	0.789	95.0
Our study	0.847	0.944	97.539

From the obtained result in Table 7, AUC(ROC) is increased by 2.53 percent with help of data processing technique, feature creation and hyperparameter tuning. The model was successfully deployed onto cloud service for remote use.

Conclusion

In this work, Diabetes prediction was achieved using a proposed classifier. Pre-processing played a crucial role in improving the performance of the classifier. The quality of data was improved with the help of techniques like outlier rejection, feature creation, standardisation and filling missing values. The correlation matrix of processed data suggests that Insulin along with glucose and skin thickness plays an important role in determining the presence of diabetes in humans. Other factors such as Diabetes Pedigree Function, Blood Pressure, Pregnancies, BMI and age also contribute to the prediction of diabetes. The optimised parameter of the ML model can improve the performance of the classifier, which was achieved with help of grid search. Table 7 demonstrates that the proposed ML model outperformed the recent proposed work on AUC(ROC) with a percentage of 97.539. Cloud-based integration of model and user-friendly UI on the web finds applications in medical sectors to predict various diseases.

Future Scope:

The above model can be extended to various disease predictions with suitable parameter extraction. The replacement of missing attributes can find an alternative way of substituting more meaningful values instead of median values. Doctor's advice can be taken into consideration for the selection of the best performance matrix and hence help build a stronger model to diagnose a diabetic patient.

REFERENCES

[1] A. Misra, H. Gopalan, R. Jayawardena, A. P. Hills, M. Soares, A. A. Reza-Albarrán, and K. L. Ramaiya, "Diabetes in developing countries," Journal of Diabetes, vol. 11, no. 7, pp. 522-539, Mar. 2019.

[2] R. Vaishali, R. Sasikala, S. Ramasubbareddy, S. Remya, and S. Nalluri,"Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset," in Proc. International Conference on Computing Networking and Informatics, Oct. 2017, pp. 1-5.

[3] Emerging Risk Factors Collaboration and other, "Diabetes mellitus, fasting blood glucose concentration, and risk of vascular disease: a collaborative meta-analysis of 102 prospective studies," The Lancet, vol. 375, no. 9733, pp. 2215-2222, Jul. 2010.

[4] N. H. Choac, J. E. Shaw, S. Karuranga, Y. Huang, J. D. R. Fernandes, A. W. Ohlrogge, and B.Malandaa, "IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045," Diabetes Research and Clinical Practice, vol. 138, pp. 271-281, Apr. 2018.

[5] P. Saeedi, I. Petersohn, P. Salpea, B. Malanda, S. Karurangaa, N. Unwin, S. Colagiuri, L. Guariguata, A. A. Motala, K. Ogurtsova, J. E. Shaw, D. Bright, R. Williams, and IDF Diabetes Atlas Committee, "Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation," Diabetes Research and Clinical Practice, vol. 157, pp. 107843, Nov. 2019.

[6] M. Maniruzzaman, M. J. Rahman, M. A. M. Hasan, H. S. Suri, M. M. Abedin, A. El-Baz, and J. S. Suri, "Accurate diabetes risk stratification using machine learning: role of missing value and outliers," Journal of Medical Systems, vol. 42, no. 5, pp. 92, May 2018.

[7] G. J. McLachlan, "Discriminant analysis and statistical pattern recognition," Journal of the Royal Statistical Society: Series A (Statistics in Society), vol. 168, no. 3, pp. 635-636, Jun. 2005.

[8] S. B. Belhouari and A. Bermak, "Gaussian process for nonstationary time series prediction," Computational Statistics & Data Analysis, vol. 47, no. 4, pp. 705-712, Feb. 2004.

[9] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, Oct. 2001.

[10] G. I. Webb, J. R. Boughton, and Zhihai Wang, "Not So Naive Bayes: Aggregating one-dependence estimators," Machine learning, vol. 58, no. 1, pp. 5-24, Jan. 2005.

[11] C. Cortes and V. Vapnik , "Support-vector networks," Machine Learning, vol. 20, pp. 237-297,

[12] T. BP and H. WH, "A multivariate logistic regression equation to screen for diabetes: development and validation," Diabetes Care, vol. 25, no. 11, pp. 1999-2003, Nov. 2002.

[13] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," IEEE Transactions on Electronic Computers, vol. 14, no. 3, pp. 326-334, Jun. 1965.Sep. 1995.

[14] D. Sisodia and D. S. Sisodia, "Prediction of diabetes using classification algorithms," Procedia Computer Science, vol. 132, pp. 1578-1585, Jan. 2018.