Analysis of Classifiers for Prediction of Type II Diabetes Mellitus

Rahul Barhate

Department of Information Technology
MITCOE, Pune

rahulbarhate97@gmail.com

Dr. Pradnya Kulkarni

Department of Information Technology,
(MITCOE) MIT World Peace University, India
Federation University, Australia
pradnya.kulkarni@mitcoe.edu.in

Abstract— **Diabetes mellitus is a chronic disease and a health challenge all over the world. As per the International Diabetes Federation, 451 million people have diabetes worldwide and this number is expected to rise up to 693 million people by 2045. It has been shown that 80% of the complications arising from type II Diabetes can be prevented or delayed by early identification of people at risk. Diabetes is difficult to diagnose in the early stages as the symptoms of the disease grow subtly and gradually. Many of the cases involve the patient being undiagnosed until they are admitted for a heart attack or begin to lose their sight. This paper analyzes the different classification algorithms based on a patient's health history to aid doctors identify the presence as well as promote early diagnosis and treatment. The experiments were conducted on Pima Indian Diabetes data set. Various classifiers used include K Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine and Neural network. Results demonstrate that random forests performed well on the data set giving an accuracy of 79.7%.**

Keywords— Diabetes Mellitus, Bioinformatics, Medical Diagnosis, Machine Learning, Classification.

I. INTRODUCTION

Diabetes Mellitus, also known as Diabetes, is a group of metabolic diseases characterized by hyper glycaemia (increase in the blood sugar levels) [1]. One of the reason for this condition is insulin deficiency, where enough insulin cannot be produced by the beta cells in the pancreas. This condition is known as type I diabetes. The other and most common reason is insulin resistance as the body's cells do not respond to insulin produced. This is known as type II diabetes [2] which accounts for 90% of all the diabetes cases. In 2015 alone, diabetes was the direct cause of 1.6 million deaths globally [16]. As per the International Diabetes Federation, 451 million people have diabetes worldwide and this number is expected to rise up to 693 million people by 2045 [18].

Diabetes has been associated with long-term damage, dysfunction, and failure of different organs, especially the eyes, kidneys, nerves, heart, and blood vessels [1]. It is responsible for over one million limb amputations and is the leading cause of blindness and visual impairment in adults in developed countries. It has also been linked with escalated risk of macrovascular complications, where people suffering from diabetes are two to four times more likely to get cardiovascular diseases compared to people without diabetes [5].

Early detection of diabetes would be of great value as at least 50% of the people and in some of the countries 80% of the people are completely unware of their condition and would remain unaware until complications occur. [2], [4]. According to recent studies it is seen that about 80% of type II diabetes complications can be prevented or delayed subject to early identification and intervention among the people at risk [2], [4].

Machine learning aims at designing algorithms that can recognize complex patterns and make intelligent decisions based on the input data [3]. The potential of machine learning algorithms to extract meaningful relationship within a data set can be used in clinical scenarios for diagnosis, treatment and predicting the outcome for several diseases [10]. These algorithms support the clinicians in their everyday duties and assist with tasks that rely on manipulation of data and knowledge. In this way, machine learning puts an arrow in the quiver of clinical decision making. The major focus of many of the ML systems is on eliminating the need of human intuition during data analysis while the others adopt a unified approach between humans and machines. The system designer needs to specify how the data is to be represented and which mechanism will be used to explore hidden patterns in the data. Hence, human intuition cannot be eliminated completely.

Several machine learning methods have been proposed for the diagnosis and management of diabetes. These techniques have the ability to revolutionize the diabetes risk prediction with the help of advanced computational methods and the availability of a data set. The classification techniques would help clinicians to make better decisions about the status of the disease among the probable patients. This paper aims at analyzing different classification algorithms used to predict the onset of diabetes.

The paper is organized as follows. Section II describes the related work done in this field. The experimental methodology is presented in Section III. This section talks about the data set and the data cleaning method used. A brief description of classifiers is put forth. It is followed by results and discussions in Section IV. Section V concludes the paper with future scope.

II. RELATED WORK

In medical diagnosis handling missing data values is of utmost importance especially when the size of the data set is small. As proposed in [8], MICE imputations have a number of advantages in handling missing data values over methods such as complete case analysis [14], single imputations [15] and maximum likelihood methods [15].

Different classification and clustering algorithms have been used for predicting diabetes. The earliest work on the Pima Indian data set can be traced back to 1988, when Smith, Everhart

[11] have performed an evaluation using a neural network model - ADAP for forecasting the onset of diabetes mellitus in the Pima Indian population. They state that neural nets provide great results when the sample size is limited and the underlying relationship involves complex intercorrelations and interactions among a number of variables. A comparative study on the Pima Indian Diabetes was performed in [24] by using a multilayer neural network trained by LM algorithm and probabilistic neural network. It was seen that neural networks could successfully predict diabetes better than the other conventional classification methods presented in [25], [26].

Machine learning methods such as non-parametric tree-based methods and support vector machines build a robust model with lower bias. Support Vector Machines were used to diagnose diabetes on Pima Indian data set in [5], [13]. Barakat have used a module which turns the black box model of an SVM into an intelligible representation of the SVM's diagnostic (classification) decision. Their results show that intelligible SVM is a promising tool for the prediction of diabetes. Asma A. AlJarullah [27] has used decision tree with an accuracy of 78.17%. In their work, Evanthia E. Tripoliti, Dimitrios I. Fotiadis [21] have presented a method of automated diagnosis of a disease based on the improved random forest algorithm. They have also addressed the issue of optimum number of base classifiers composing the random forest.

III. METHODS AND MATERIALS

A. Data set Description: -

The data set used in the experiment is the PIMA Indian diabetes data set which was made public by UCI. The patients in the data set are females who are at least twenty-one years old of Pima Indian heritage living near Phoenix, Arizona, USA. The data set consists of 768 records each record having 8 features and labels indicating whether or not each patient was diagnosed with diabetes. The various features taken under consideration are as follows:

Table 1. Feature Description

Attribute	**Description**
Pregnancies	Number of pregnancies.
Glucose	Plasma glucose concentration (2 hours in an oral glucose tolerance test).
Blood Pressure	Diastolic blood pressure (mm Hg).
Skin Thickness	Triceps skin fold thickness (mm).
Insulin	2-Hour serum insulin (mu U/ml).
BMI	Body mass index (weight in kg/ (height in m) ^2).
Diabetes Pedigree Function	Diabetes pedigree function.
Age	Age (years).
Outcome	Class variable 0 – Non-diabetic 1 – Diabetic

The Diabetes Pedigree function provides some data about the diabetes history among the patient's relatives and the genetic relationship between that relative and the patient. This measure of genetic influence gives an insight about the hereditary risk a patient might have regarding the onset of diabetes.

B. Data Preparation: -

The data set contains some missing values that have been replaced by zeros. The number of missing values for each of the attributes are as follows:

Table 2. Missing values for the features

Attribute	**No of missing values**
Pregnancies	110
Glucose	5
Blood Pressure	35
Skin Thickness	227
Insulin	374
BMI	11
Diabetes pedigree function	0
Age	0

Except for pregnancies, all the other zero values are treated as missing values. Various approaches exist for handling missing data. The first and simplest approach is to delete all the records with missing values for the variable which is under consideration. This technique leads to loss of potentially valuable information about the patients whose records are deleted. The second approach is to replace all the missing values with mean/median of the data value of the variable under consideration. This would introduce bias in the data set and may lead to poor classification. As the size of the data set is small, it is essential to obtain as much information from it as possible and hence it becomes crucial not to delete the entire observations (rows) or variables (columns) containing missing values. Taking this into consideration, two approaches have been taken to deal with missing values. They are as follows:

1) Replacing the missing data with median values on smaller number of missing values. (Records having missing values for Glucose, Blood Pressure and BMI)
2) Replacing the missing data with prediction values. (Multiple Imputations).
   (Records having missing values for Skin Thickness and Insulin)

The following figure 1 illustrates the distributions of the Skin thickness and Insulin levels before and after performing MICE imputations.

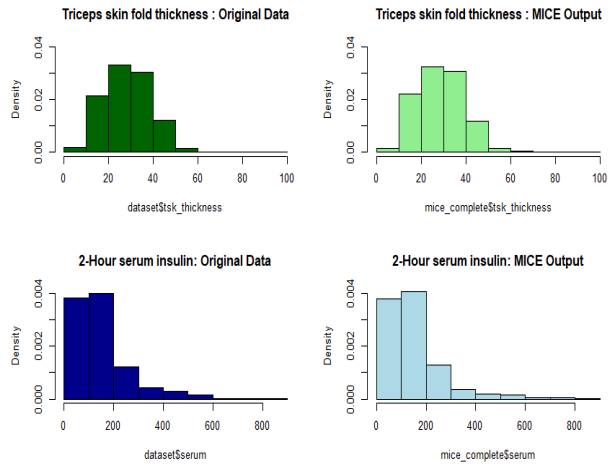

Fig. 1. MICE Imputations

Compared to the original distributions, the distributions after imputation haven't changed significantly. Hence, a cleaned data set is available.

Multiple Imputation by Chained Equations (MICE)

MICE has emerged as a principle method for handling missing data. It has properties that make it particularly useful for large imputations and advances in software development. MICE, sometimes also called as 'Sequential Regression Multiple Imputation' creates multiple imputations that can account for statistical uncertainty in imputations as against single imputations. This approach is very flexible and can handle varying type of variables (e.g. continuous and binary) as well as survey skip patterns.

Missing data is a very common issue in medical research. In certain situations where the missingness is less than 5% and it is completely random and does not depend on observed or unobserved values, complete case analysis is a reasonable approach to handle missing values [6][7]. Although complete case analysis might seem easy to implement it tends to rely on stronger missing data assumptions than multiple imputations and thus can lead to bias estimates [6]. Single imputations like mean imputation may be an improvement but they do not account for uncertainty in the imputations. After the completion of imputations, the analysis proceeds in a manner which suggests that the imputed values were known and true values rather than imputed values. This eventually leads to overly precise results and possess the potential to come to incorrect conclusions. Sometimes maximum likelihood methods are a feasible solution to handle missing values [6], however these methods are applicable only for certain types of models such as longitudinal or structural equation models and can generally run using special software.

Multiple imputations have an advantage over the above-mentioned methods [8]. It involves filling in the missing values multiple times in order to create a "complete" data set. Elaborated in [9], the missing values are imputed based on the observed values of a particular individual and the relations observed on the data for other individuals assuming that the observation variables have been included in the imputation method.

As described in [8], the chained equation process is broken down into the following steps:

Step 1: A mean imputation is performed for every missing value in the data set.

Step 2: The mean imputations for one of the variable ("var") are then set back to missing.

Step 3: The observed values from the variable "var" in Step 2 are regressed on the other variables in the imputation model, which may or may not consist of all the other variables in the data set.
i.e. "var" is the dependent and all the other variables are independent variables in the regression model. These regression models work in the same way like they do while performing (e.g.) linear or logistic regression models outside the context of imputing missing data.

Step 4: The missing values for "var" are then replaced with predictions (imputations) from the regression model. When "var" is subsequently used as an independent variable in the regression models for other variables, both the observed and the imputed values are used.

Step 5: Steps 2 through 4 are then repeated for each variable that have missing values.
The cycling through each of the variables constitutes one iteration or "cycle." At the end of one cycle all of the missing values have been replaced with predictions from regressions that reflect the relationships observed in the data.

Step 6: Steps 2 through 4 are repeated for the number of cycles, with imputations being updated at every cycle. The number of cycles to be performed are specified by the user. At the end of these cycles the final imputations are retained, resulting in one imputed data set.

Usually after the end of the cycles, the distribution of the parameters governing the imputations (i.e. the coefficients in the regression models) are expected to have converged so that they are stable.

Considering the PIMA Indian data set, there are 8 features in the data set among which - Skin Thickness and BMI have large amount of missing values. An MAR assumption implies that the probability of a particular variable to be missing depends only on the observed values, for example, whether someone's BMI is missing does not depend on their (unobserved) BMI. In step 1 of the MICE process, each variable would first be imputed using, e.g., mean imputation, temporarily setting any missing value equal to the mean observed value for that variable. In step 2, the imputed mean values of BMI would be set back to missing. In step 3, a linear regression of BMI predicted by other 7 variables would be run using all cases where BMI was observed to be missing. In step 4, predictions of the missing BMI values would be obtained from that regression equation and imputed. At this point, BMI does not have any missingness. Steps 2–4 would

then be repeated for the Skin thickness variable. The originally missing values of Skin Thickness would be set back to missing and a linear regression of income predicted by the other 7 variables would be run using all cases with Skin thickness observed; imputations (predictions) would be obtained from that regression equation for the missing income values. This entire process of iterating through the 2 variables would be repeated until convergence. The observed data and the final set of imputed values would then constitute one "complete" data set.

Instead of linear or logistic regression other machine learning models can be used. In this experiment, a Random Forest model has been used.

C. Exploratory data analysis: -

Upon examination of the distribution of the class values, it is seen that there are 500 negative instances (65.1%) and 258 positive instances (34.9%). The figure 2 illustrates the number of diabetic and non-diabetic records in the data set.

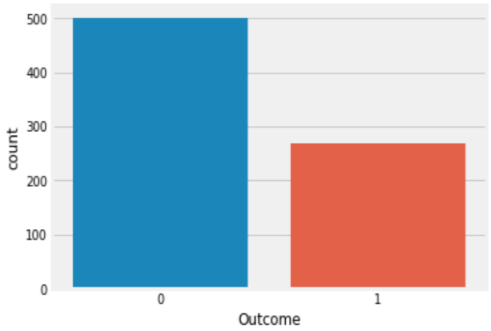

Fig. 2. Diabetic and Non-diabetic patients.

Histograms representing a single variable and are used to visualize the shape of the distribution of the particular variable. This distribution conveys how often a value occurs. The histograms for all the features in the data set is as represented in the following figure 3.

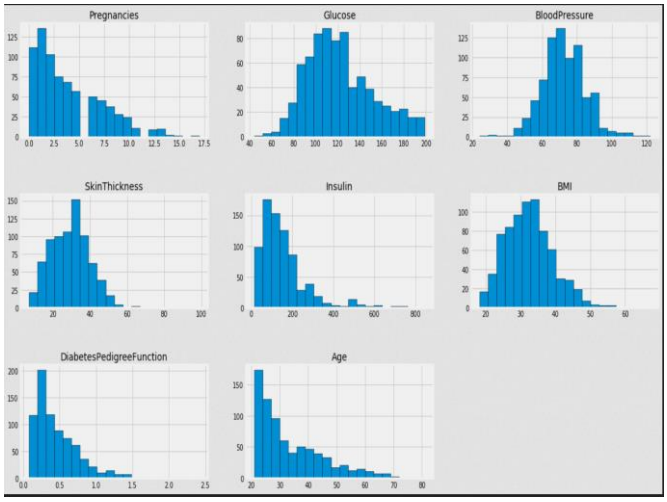

Fig. 3. Histogram of features for all the records in the data set.

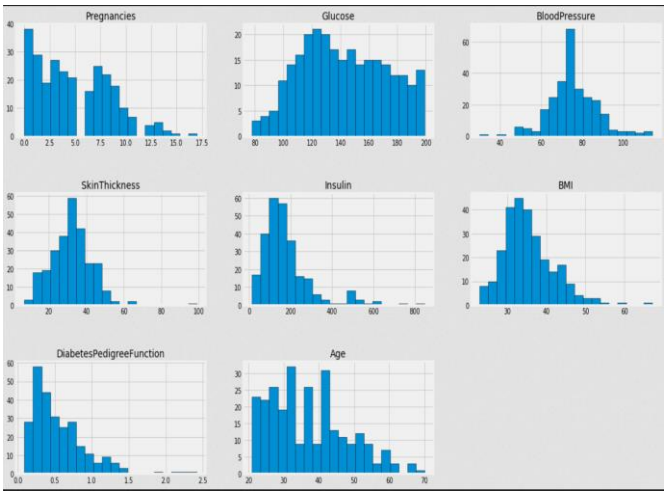

Fig. 4. Histogram of features for diabetic patients in the data set.

Similarly, the histograms for the features of the patients diagnosed with diabetes has been illustrated in figure 4. Reviewing the histograms, it is seen that some of the attributes such as Glucose, Pressure, Skin Thickness and BMI look normally distributed whereas features such as Pregnancies, Insulin, Diabetes pedigree function and age are exponentially distributed. Probably age should have a normal distribution but it seems that the constraints on the data collection might have skewed the distribution.

The distribution of the features in the data set can be represented by a pair plot.

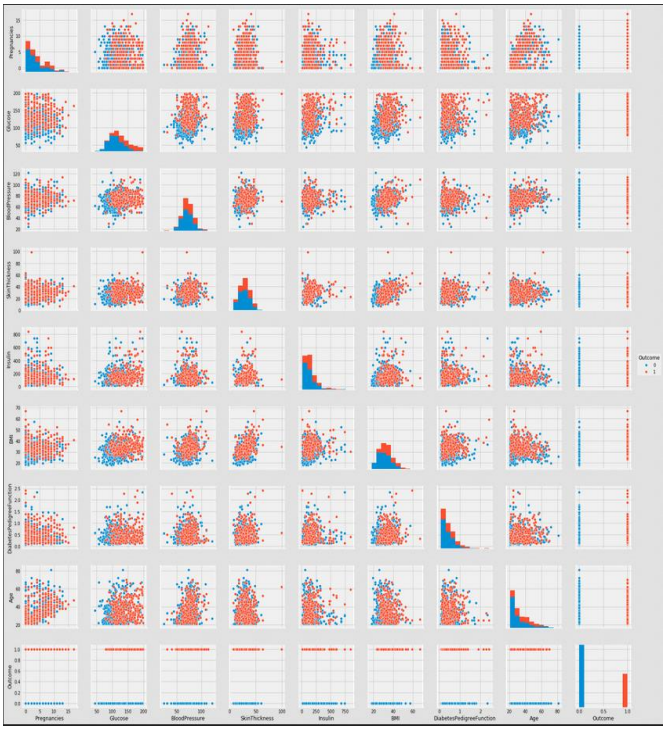

Fig. 5. Pair plot.

From the pair plot it is seen that there is no obvious relationship between age and the onset of diabetes or diabetes pedigree function and the onset of diabetes. Apart from these, no two other

attributes are clearly able to separate the two outcome-class instances.

D. Description of Classifiers: -

1. K-Nearest Neighbors: -

KNN is a supervised learning algorithm which takes a set of labelled points and learns how to label other points in the problem space. Whenever a new point is to be labelled, it looks at the closest labelled points from the new point. These points can be considered as the neighbors of the new point and allows these neighbors to vote. The label that most of the neighbors have is assigned as the label of the new point.

2. Logistic Regression: -

It is a statistical method used for analyzing a data set which consists of one or more independent variables that predict an outcome. A dichotomous variable (takes only two possible values) is used to measure the outcome. Logistic regression finds the best fitting model that describe the relationship between the dichotomous characteristic of interest (dependent variable) which is response or outcome variable and a set of independent variables.

3. Decision Tree

Decision tree builds classification or regression models in the form of a tree structure. The data set is broken down into smaller and smaller subsets and at the same time an associated decision tree is developed incrementally. Finally, this results in a tree with decision nodes and leaf nodes. The root node is the topmost decision node in a tree. A decision node has two or more branches and a leaf node represents a classification or decision.

4. Random Forest

Random Forest algorithm is an ensemble learning method that operates by constructing huge amount of independent decision trees and outputting the class that is either the mean prediction (regression) or mode prediction (classification) of the individual trees. They are used to lower the variance keeping the bias constant. Hence, they overcome decision tree's habit of over fitting.

5. Gradient Boosting

Gradient Boosting is a method which combines weak learners like the decision trees and ensembles them with the help of weighted majority vote to predict the class. Initially it starts with constructing a small tree and builds additive tree models sequentially. The subsequent predictors learn from the mistakes of the earlier predictors and hence observations have an unequal probability to appear in the subsequent models.

6. Support Vector Machines (SVM)

SVM is a discriminative classifier wherein each data point is plotted as a point in n-dimensional space (n is the number of features) and the value of the feature indicates the value of a particular coordinate. The algorithm constructs one or more hyperplane (decision boundary) in the feature space that differentiates the data points into different classes.

7. Neural Network

A neural network consists of units (neurons), arranged in series of layers, which convert an input vector into some output. Each neuron takes an input, applies a function to it and passes the output on to the next layer. A multi-layered perceptron is defined to be feedforward i.e. a unit feeds it output to all the units in the successive layer, but there is no feedback to the previous layer. Weights are applied to the output which is passed on from one unit to another. These weights represent the strength of the connection between two neurons i.e. if the weight of the connection between unit A and unit B has a greater magnitude than any other connection to unit B then it means that A has a greater influence in increasing or decreasing the activation levels of B. These weights are then tuned in the training phase so that the neural network adopts to the problem at hand.

IV. RESULT AND DISCUSSIONS

All the experiments are performed using the Scikit-learn library [17] in Python. The data set is split into 70% for training and 30% for testing. For each of the classifier, optimization is done by tuning the associated model parameters. This is accomplished by training the model on the training set. The accuracy of the training set is a natural metric used to evaluate the efficiency of model in a classification model.

Experiment 1: KNN

In figure 6, the y-axis represents the training and test accuracy against the number of neighbors on the x-axis.

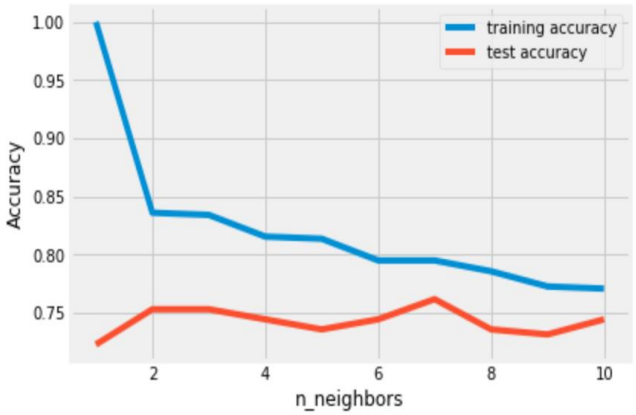

Fig. 6. Accuracy vs Number of neighbors in K-nearest neighbors

With a single neighbor the training accuracy is perfect but the test accuracy drops which indicates that the model is overfitting and is too complex. On the other hand, with 10 neighbors the performance is even worse and the model is too simple. As the number of neighbors are increased from a single neighbor the training accuracy starts dropping. The optimum level of test accuracy is seen when the number of neighbors is equal to 7. Hence, we have chosen 7 neighbors to evaluate the training and

testing accuracy of the K-nearest neighbors algorithm. The training and testing accuracy with n = 7 is 80% and 76% respectively.

Experiment 2: Logistic Regression

The default value of C = 1 (inverse of regularization strength) provides with 76.9% accuracy on the training and 77.5% accuracy on the test set. Using C = 0.01 results in 69.3% and 71.4% accuracy on the training and test set respectively. Using C = 100 results in a little bit if higher accuracy on the training set and a little bit lower accuracy on the test set. This confirms that a complex model with less regularization may not generalize better than the default parameter values. Therefore, the default value of C = 1 is chosen.

Experiment 3: Decision Tree

Using the best split, the training accuracy of the decision tree is 100% while the test set accuracy is 70.6% which is very poor. This indicates that the tree is overfitting and not generalizing well to the new data. A solution to this problem is applying pre-pruning to the tree [25].
Overfitting has been decreased by limiting the maximum depth of the tree to 4. This lead to lower accuracy on the training set (80.6%) but an improvement in the test set (75.8%).

Feature importance rates how significant a particular feature is for the decision that the tree takes. For every feature it is a number between 0 and 1, where 0 means that the feature was not used at all and 1 means that the feature perfectly predicts the target. The feature importance's for all the variables always sum up to 1.

Figure 7 illustrates the feature importance for each of the features in the decision tree.

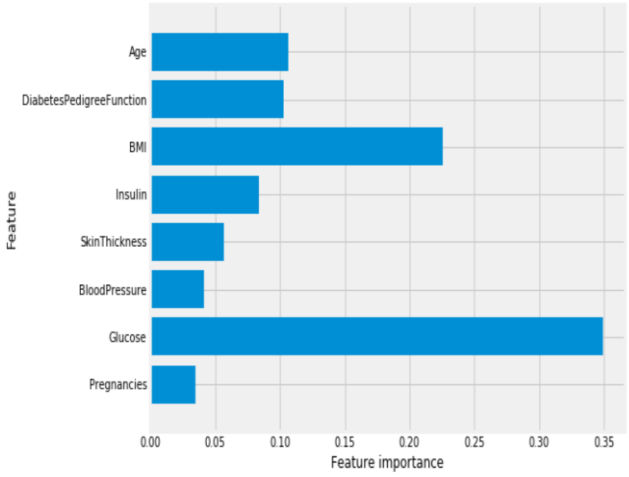

Fig. 7. Feature importance in decision tree.

Here, it is seen that Glucose is the most important feature.

Experiment 4: Random Forest

A random forest consisting of 100 trees achieves an accuracy of 79.7% on the test set. However, by adjusting the maximum depth of the tree the training and test accuracy decrease. Hence, the default parameters are chosen for the decision tree.
Like the single decision tree, the random forest gives a lot of

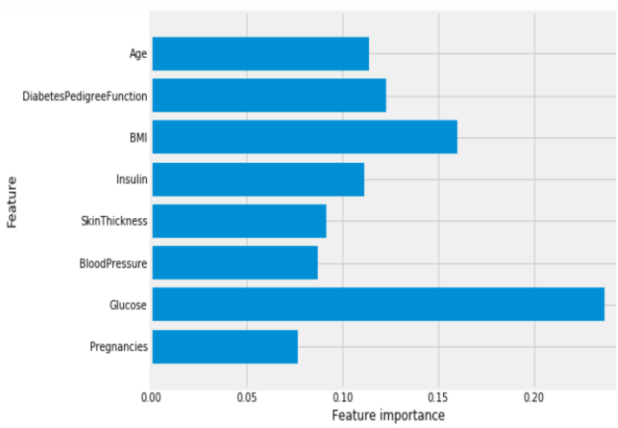

Fig. 8. Feature importance in random forest

importance to the 'Glucose' feature, followed by the 'BMI' feature. The randomness in building the forest compels the algorithm to consider many possible explanations. Due to this, the random forest captures a much broader picture of the data than a single decision tree.

Experiment 5: Gradient Boosting

Without optimizing the parameters gradient boosting achieves and accuracy of 91.7% on the training set and 79.2% on the test set. It is likely that the model is overfitting. Overfitting in gradient boosting can be reduced by pre-pruning by limiting the maximum depth or lowering the learning rate. By limiting the maximum depth of the tree to 3 an improvement is seen with 93.5% and 79.2% on train and test while optimal performance is achieved when the C = 0.01 with 80.4% and 75.8% on the train and test set respectively. Both the methods of decreasing the model complexities reduced the training set accuracy and didn't increase the generalization performance of the test set. Figure 9 shows the feature importance of the gradient boosting method. It is observed that like random forest gradient boosting give importance to all the features.

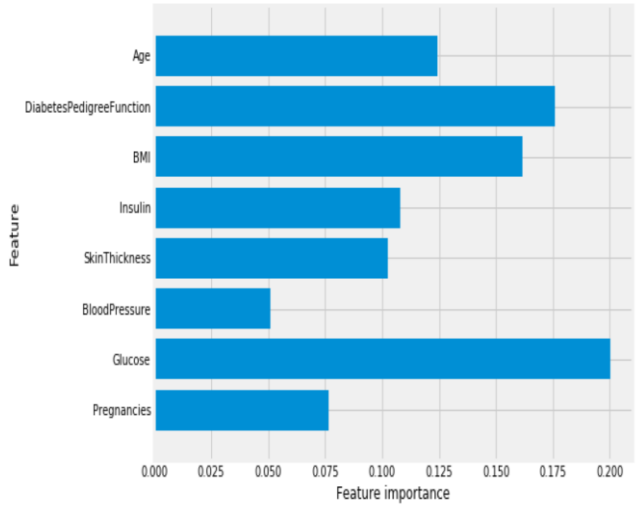

Fig. 9. Feature importance in gradient boosting.

Experiment 6: Support Vector Machine

The model overfits quite significantly with a perfect score on training set and a poor 65% on the test set. SVM requires all the features to vary on a similar scale. Data rescaling is done so that all the features are approximately on the same scale. For this purpose, a Min-Max Scaler [21] is used. Huge difference in the accuracies can be seen after scaling the data. Now, the model is underfitting where the training and test set performance is very similar (77%). Hence, two approaches are adopted in order to fit a complex model.

1) Increasing C
2) Increasing gamma

Increasing the C (C=1000) helps in improving the model to achieve an accuracy of 79.9% and 77.9%.

Experiment 7: Neural Networks

Due to scaling of the data, the accuracy of the multilayered perceptron isn't as good as the other models. Ideally, the deep learning models expect that the input features vary in a similar way having a mean of 0 and a variance of 1 [23]. Hence, keeping this in mind standardization of the data is done. Standard Scaler [20] is used for this purpose. An accuracy of 81.9% on the training set and 78.4% on the test set is achieved. Increasing the default 100 iterations to 1000 iterations the neural net improves on the training accuracy but the test accuracy is reduced to 77.9%. Increasing the alpha parameter to add string regularization of the weights both the training and test accuracy drop. Hence, the neural net after scaling provided the optimum results.

The table 3 summarizes the accuracies of all the different algorithms.

Table 3. Algorithm Accuracies

Sr. No.	Algorithm	Training Accuracy	Test Accuracy
1	K-Nearest Neighbours	80%	76%
2	Logistic Regression	76.9%	77.5%
3	Decision Tree	80.6%	75.8%
4	Random Forest	100%	79.7%
5	Gradient Boosting	93.5%	79.2%
6	Support Vector Machine	79.9%	77.9%
7	Neural Network	81.9%	78.4%

V. CONCLUSION AND FUTURE WORK

In this paper we have tried to analyze various classification algorithms for prediction of Type II Diabetes Mellitus. The missing values in the PIMA Indian Diabetes data set are fixed with the help of MICE imputations. Experiments were performed on the thus cleaned PIMA Indian data set using different classification algorithms. The classification models were tuned by varying the different parameters for optimum performance. Various advantages and disadvantages of each model are noted. From the experiments conducted it was seen that random forest performed well on the Pima Indian data set giving an accuracy of 79.7%. However, it is seen that there was no significant difference in the accuracies provided by the various classifiers. The future extension is aimed at ensemble method that combines several base models for further improvement in predictive accuracy.

REFERENCES

[1] American Diabetes Association. "Diagnosis and Classification of Diabetes Mellitus." *Diabetes Care* 33. Suppl 1 (2010): S62–S69. *PMC*. Web. 3 May 2018.

[2] International Diabetes Federation, *Diabetes Atlas*, 3rd ed. Brussels, Belgium: International Diabetes Federation, 2007.

[3] Sarwar, Abid & Sharma, Vinod. (2014). Comparative analysis of machine learning techniques in prognosis of type II diabetes. AI & SOCIETY. 29. 10.1007/s00146-013-0456-0.

[4] M. Franciosi, G. D. Berardis, M. C. E. Rossi, and M. Sacco, "Use of the diabetes risk score for opportunistic screening and impaired glucose tolerance," *Diabetes Care*, vol. 28, no. 5, pp. 1187–1193, 2005.

[5] N. Barakat, A. P. Bradley and M. N. H. Barakat, "Intelligible Support Vector Machines for Diagnosis of Diabetes Mellitus," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 4, pp. 1114-1120, July 2010. doi: 10.1109/TITB.2009.2039485.

[6] Graham JW. Missing data analysis: making it work in the real world. Annual Review of Psychology. 2009;60:549–576.

[7] Schafer JL. Multiple imputation: a primer. Statistical Methods in Medical Research. 1999;8:3–15.

[8] Azur, Melissa J. et al. "Multiple Imputation by Chained Equations: What Is It and How Does It Work?" *International journal of methods in psychiatric research* 20.1 (2011): 40–49. *PMC*. Web. 5 June 2018.

[9] Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods, 7*(2), 147-177.

[10] Ramesh, A. N. et al. "Artificial Intelligence in Medicine." *Annals of the Royal College of Surgeons of England* 86.5 (2004): 334–338. *PMC*. Web. 6 June 2018.

[11] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., & Johannes, R. S. (1988). Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 261–265.

[12] Lim, TS., Loh, WY. & Shih, YS. Machine Learning (2000) 40: 203. https://doi.org/10.1023/A:1007608224229

[13] S. Karatsiolis and C. N. Schizas, "Region based Support Vector Machine algorithm for medical diagnosis on Pima Indian Diabetes dataset," *2012 IEEE 12th International Conference on Bioinformatics & Bioengineering*

(BIBE), Larnaca, 2012, pp. 139-144. doi: 10.1109/BIBE.2012.6399663

[14] Ian R. White, John B. Carlin, “Bias and efficiency of multiple imputation compared to complete-case analysis for missing covariate values”, S*tatistics in Medicine (2010), Vol. 29 Issue 28*, page 2920-2931.

[15] Alan C. Acock, “Working With Missing Values”, *Journal of Marriage and Family (2005), Vol. 67 Issue 4,* pp. 1012-1028

[16] World Health Organization (2017): http://www.who.int/news-room/fact-sheets/detail/diabetes.

[17] Scikit-learn, (2018): http://scikit-learn.org/stable/index.html

[18] Cho NH, Shaw JE, Karuranga S, Huang Y, da Rocha Fernandes JD, Ohlrogge AW, Malanda B, “*IDF Diabetes Atlas: Global estimates of diabetes prevalence for 2017 and projections for 2045”, Diabetes Res Clin Pract. 2018 Apr;138:271-281.*

[19] Standrad Scaler: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[20] Min-Max Scaler: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[21] E. E. Tripoliti, D. I. Fotiadis and G. Manis, "Automated Diagnosis of Diseases Based on Classification: Dynamic Determination of the Number of Trees in Random Forests Algorithm," in *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 615-622, July 2012.
doi: 10.1109/TITB.2011.2175938.

[22] G. Eason, B. Noble, and I.N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955. (*references*)

[23] LeCun Y., Bottou L., Orr G.B., Müller K.R. (1998) Efficient BackProp. In: Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 1524. Springer, Berlin, Heidelberg

[24] Hasan Temurtas, Nejat Yumusak, Feyzullah Temurtas, "A comparative study on diabetes disease diagnosis using neural networks", Expert Systems with Applications,Volume 36, Issue 4, 2009,Pages 8610-8615, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2008.10.032..

[25] Carpenter, G. A., & Markuzon, N. (1998). ARTMAP-IC and medical diagnosis: Instance counting and inconsistent cases. Neural Networks, 11, 323–336.

[26] Deng, D., & Kasabov, N. (2001). On-line pattern analysis by evolving self-organizing maps. In Proceedings of the fifth biannual conference on artificial neural networks and expert systems (ANNES) (pp. 46–51).

[27] A. A. Al Jarullah, "Decision tree discovery for the diagnosis of type II diabetes," 2011 International Conference on Innovations in Information Technology, Abu Dhabi, 2011, pp. 303-307. doi: 10.1109/INNOVATIONS.2011.5893838.