Prediction of 'Type1 Diabetes' using Random Forest

Md. Shahadat Anik Sheikh
Department of Computer
Science & Engineering
East West University
2019-1-60-068@std.ewubd.edu

Talha Zubaer Siddique Alquraishy
Department of Computer
Science & Engineering
East West University
2019-1-60-116@std.ewubd.edu

A. M. Feroz Ehses Shishir
Department of Computer
Science & Engineering
East West University
2019-1-60-108@std.ewubd.edu

Abstract**:** In this project we focused on Type1 Diabetes. Type1 Diabetes is an autoimmune disease. Doctors do not know the exact reason of this type of diabetics the immune system erroneously targets and kills insulin-producing beta cells in the pancreas. Genes may have a role in certain persons. It's also conceivable that a virus triggers the immune system's onslaught. [1] We tried to figure out the impact of some attributes like age, family history of who are affected etc. on type 1 diabetes. We have worked with different algorithms and tried to find which approach gives a better result.

I. INTRODUCTION

Diabetes mellitus commonly known as diabetes, has become a common disease and is increasing among people. There are some common symptoms of diabetes which are increased hunger, increased thrust, weight loss, frequent urination etc. There are different types of diabetes like type1, type2, prediabetes and gestational diabetes from among these the cause of diabetes type1 is still unknown to doctors and symptoms can appear suddenly. So, this paper tries to figure out some of the aspects which are connected and maybe the main reason for the disease. We have implemented various models for identifying diabetes based on some aspects like Age, Gender, Height, Weight, Adequate nutrition, Standard growth rate, Antibody production of the body and previous family history. These approaches may be divided into two main categories: supervised learning and unsupervised learning algorithms. The models include Decision Tree, Naïve Bayes, Random Forest algorithms to determine the accuracy of the prediction.

A. Decision Tree

A Decision Tree is a classification technique that uses a tree structure, where each node represents a characteristic and the branch represents the value of the characteristic, at the same time as the leaves are used to represent the class. Decision Trees are generally being applied as like human thinking while making a decision. The logic behind the Decision Tree can be

easily understood as it indicates a tree-like structure. So, it is straightforward to understand.

The single top node of the Decision Tree is referred to as the root which becomes possible outcomes by spreading branches. Each of those outcomes leads to additional nodes, which branch off into other possibilities. [2]

There are three different types of nodes. They are chance nodes, decision nodes, and end nodes. A chance node represented by a circle which shows the chances of certain results. A decision node represented by using a rectangular, shows a decision to be made, and an end node indicates the final results of a decision path. However, in our Decision Tree every node is represented by rectangular shape.

The Decision Tree is used in many real-life sectors, which include engineering, civil planning, regulation, and business. this can compare potential growth for organizations, which primarily based on previous historic facts or data. In several fields like logistics managements, strategic managements are planned using Decision Tree. Decision Tree enables finding prospective customers in business purposes. Decision Trees are easy to study, interpret and put together. Also, much less data is needed when a variable is created once.

B. Random Forest Algorithm

Random Forest algorithm is also called as Random Decision Forest (RDF), that is used for Classification, Regression and other tasks that are performed by constructing a forest. The "Forest" it builds, is an ensemble of Decision Trees. [3]

This Random Forest Algorithm is primarily based on supervised learning. It is flexible and easy to apply machine learning algorithm that produces an extremely good end result most of the time, even without hyper-parameter tuning. This algorithm offers us higher accuracy, when compared with all other current systems. Farther more, this is most usually used algorithm, due to its simplicity and diversity. However, the major benefit of this algorithm is that it can be used for both class and Regression.

Overfitting issues can be solved through using Random Forest Algorithm, if there's a sufficient number of trees. Hyperparameters, it uses often produce a good and accurate prediction result. But more accurate prediction requires extra trees, which results the algorithm quite ineffective and slower to develop for real time situations, even though it can take care of quite a few different types feature. Despite this, the Random Forest Algorithm is broadly used in a lot of different fields like banking, the stock market, medicine and e-commerce.

C. Naïve Bayes Algorithm

Naïve Bayes is a simple probabilistic machine learning algorithm that utilizes Bayes rule together with a strong assumption that the attributes are conditionally independent, given the class. It is based on probability models that incorporate strong independence assumptions. The independence assumptions often do not have an impact on reality. Therefore, they are considered as Naïve.

Naïve Bayes a simple yet powerful algorithm. That's because there is a significant advantage with Naïve Bayes. Since it is a probabilistic model, the algorithm can be coded up easily and the predictions made really quick. Because of this, it is easily scalable and is traditionally the algorithm of choice for real-world applications that are required to respond to user's requests instantaneously [4]. But this cost us accuracy of that prediction. Despite this, the Naïve Bayes algorithm is broadly used in a lot of different fields like spam filtering in email sites, recommendation systems and so on.

II. RESEARCH METHOD

Our work has a sequential flow that was followed to achieve the result. After we select the dataset, we checked if there is any missing data in any column, as most of the columns were in string, we encoded the data; trained a portion of data and set the test dataset. Then we applied the classifiers on the dataset and compared the results.

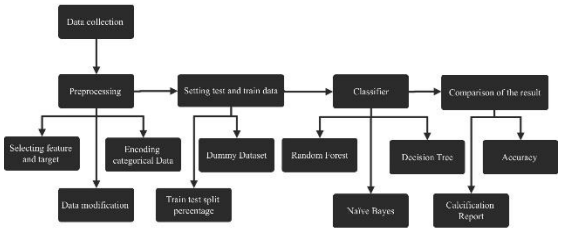

Fig: Process flow

A. Preprocessing:

We tried to process the dataset before we start working with it. We found no missing data but still some results were unknown to us, we avoided those results. Then we dropped the unnecessary attributes for our analysis (i.e. Area of Residence, Duration of disease, Education of Mother etc.). These attributes have very less significance in our analysis. After the elimination of these attributes, we have worked with 18 columns as input and set the target attribute is 'Affected'.

In our dataset many columns were in string but the classification method that we have used can only work with integer and float data. Therefore, we have converted our dataset to integers from string. But this modification increased the number of columns. However, all the columns were defined in number data type. Finally, we have split the dataset into features and target column.

The feature attributes are "Age", "Sex", "HbA1c" (glycated hemoglobin level which indicates the average blood sugar level for a period of a week or month), "Height", "Weight", "BMI", "Duration of disease, "Other diseases", "Adequate", "Nutrition", "Standardized growth-rate in infancy", "Standardized birth weight, Autoantibodies", "Impaired glucose metabolism", "Insulin taken", "Family History affected in Type 1 Diabetes", "Family History affected in Type 2 Diabetes", "Hypoglycemis", "pancreatic disease affected in child" and the target attribute is "Affected''.

B. Setting Test Dataset:

This section contains two part: The Dummy dataset and splitting of test and train data.

i) Applying the Dummy Dataset:

As some of the columns are string data and there were three to four different values in a column so, we made some extra columns to identify that data in binary. However, for each column there were several columns and at last we concat the columns to get back the previous columns in integer. To do this we have used

get_dummy algorithm from panda's data frame.

We did not use label encoder because there were more than two distinct data in some columns where the label encoder would not give us the proper result.

ii) Split data into train and test:

The Diabetes_Type1 dataset will be used as both test and trained dataset. For each algorithm we will split this dataset into 25% which implied that 25% data will be for testing purpose and the rest 75% will be for training purpose.

C. Classifier Algorithm:

For this dataset as there are many possible outputs analysis can be done using Decision Tree, Random Forest and Naïve Bayes, we will be implementing this three algorithm for target.

D. Comparison of the result:

After finishing the classifier implementation for the target variable "Affected", the result found will be compared with each other. For example, when the Decision Tree, Random Forest and Naïve Bayes are used for target variable prediction the result of these three classifiers will be compared to decide which one is better for this dataset.

III. Result Analysis and Discussion

Following steps have been used to get the result and compare between the results based on Decision Tree, Naïve Bayes and Random Forest algorithm.

Classification report: Predicting Affected in Type1 Diabetes in Decision Tree, Random Forest and Naïve Bayes

Accuracy using Decision Tree: 93.34%

Maximum Accuracy: 93.62%

Minimum Accuracy: 93.06%

Overall Accuracy: 93.34%

Output	Precision	Recall	F1-Score	Support
Affected	0.93	0.92	0.92	7103
Not Affected	0.94	0.94	0.94	9271
Accuracy			0.93	16374
Macro avg	0.93	0.93	0.93	16374
Weighted avg	0.93	0.93	0.93	16374

Table 1.1: Classification report for Decision Tree

Accuracy using Random Forest: 93.50%

Maximum Accuracy: 93.70%

Minimum Accuracy: 93.28%

Overall Accuracy: 93.50%

Output	Precision	Recall	F1-Score	Support
Affected	0.93	0.92	0.92	7095
Not Affected	0.94	0.94	0.94	9279
Accuracy			0.93	16374
Macro avg	0.93	0.93	0.93	16374
Weighted avg	0.93	0.93	0.93	16374

Table 1.2: Classification report for Random Forest

Accuracy using Naïve Bayes: 89.24%

Maximum Accuracy: 89.40%

Minimum Accuracy: 89.08%

Overall Accuracy: 89.24%

Output	Precision	Recall	F1-Score	Support
Affected	0.92	0.83	0.87	7264
Not Affected	0.88	0.94	0.91	9110
Accuracy			0.89	16374
Macro avg	0.90	0.89	0.89	16374
Weighted avg	0.89	0.89	0.89	16374

Table 1.3: Classification report for Naïve Bayes

Algorithm implementation results:

Decision Tree:

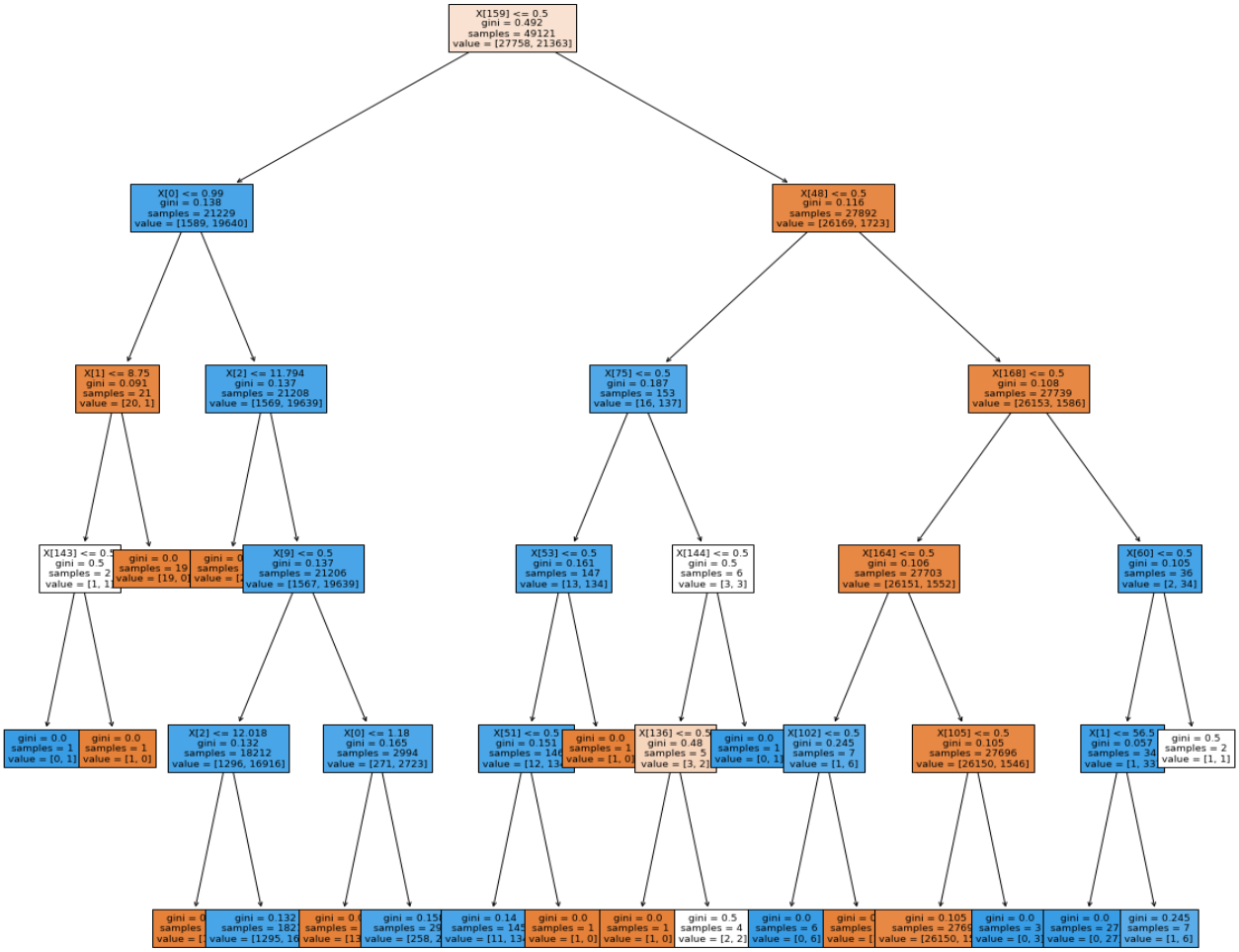

Fig 1: Decision Tree for predicting affected Diabetes Type1 patient

Confusion Matrixes:

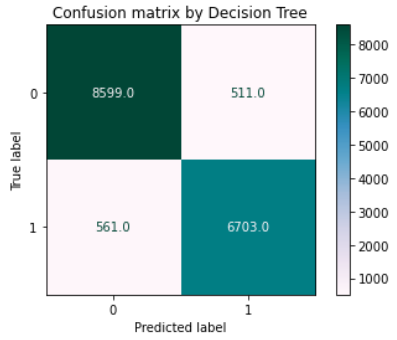

Fig 2: Confusion matrix for Decision Tree

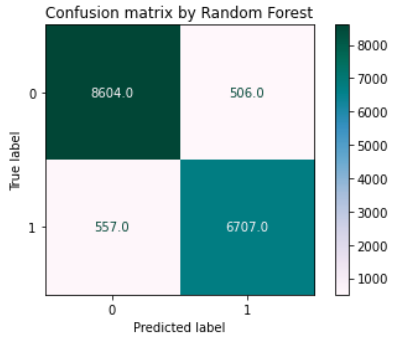

Fig 3: Confusion matrix for Random Forest

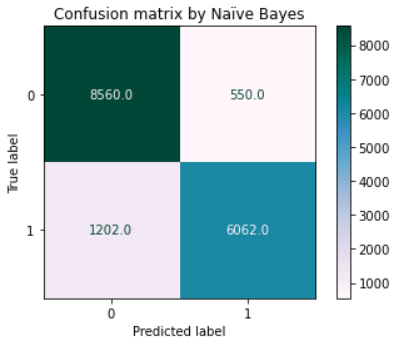

Fig 4: Confusion matrix for Naïve Bayes

Table 1.1, table 1.2 and table 1.3:

All of the table contains the affected accuracy of Diabetes Type1 for individual algorithm. It shows the main classification metrics: precision, recall and F1-score. Precision is the ability of a classifier not to label an instance positive that is actually negative. The recall value signifies what percentage of positive cases we have managed to catch. The f1 score means the percentage of positive predictions that were correct. The range for F1-score is 1 as in the best and 0 as in the worst. [5]

From the classification report we see that the overall accuracy for Decision Tree, Random Forest and Naïve Bayes sequentially 93.34%, 93.50 % and 89.24 %.

As the result shows that Random Forest is slightly better than Decision Tree and far better than Naïve Bayes. Also in theory says that, in this kind of condition or dataset Random Forest works better than any machine learning algorithm.

Figure 1:

It shows the visualization of Decision Tree.

Figure 2, figure 3 and figure 4:

In figure 2, there are 8599 True negative patients who are not affected and 6703 True positives patients who are affected that is measured correctly by Decision Tree. Here, 511 False positive patients who are not affected actually but they are affected as per prediction and 561 False negatives who are not affected according to prediction but do affected actually.

In figure 3, there are 8604 True negative patients who are not affected and 6707 True positives patients who are affected that is measured correctly by Random Forest. Here, 506 False positive patients who are not affected actually but they are affected as per prediction and 557 False negatives who are not affected according to prediction but do affected actually.

In figure 4, there are 8560 True negative patients who are not affected and 6062 True positives patients who are affected that is measured correctly by Naïve Bayes. Here, 550 False positive patients who are not affected actually but they are affected as per prediction and 1202 False negatives who are not affected according to prediction but do affected actually.

IV. Conclusion

Decision Tree has low bias but high variance. Its training accuracy is more but testing accuracy is a bit low. Whereas Random Forest gives a better performance than Decision Tree.

Naïve Bayes is a compact model which cannot model patterns in a data so, it is typically not as accurate as Decision Tree or Random Forest. Support vector machine works well with small dataset as our dataset is huge it could not give us a satisfactory result.

Random Forest is a bit ensemble method which works with dynamically sampled data which are subsets of samples and create many trees then when the test data set is provided it counts majority decision for either it is affected or not. Model complexity of Random Forest is high as it deals with so many Decision Trees and its bias is also high but variance is low so it

gives better result than Decision Tree. So out of these algorithms, Random Forest is the optimal one which give us the best result.

References

[1] S. Watson, "Everything You Need to Know About Diabetes," 26 02 2020. [Online]. Available: https://www.healthline.com/health/diabetes.

[2] "What is a Decision Tree Diagram," [Online]. Available: https://www.lucidchart.com/pages/decision-tree.

[3] N. Donges, "A Complete Guide to the Random Forest Algorithm," 22 July 2021. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm. [Accessed 10 09 2021].

[4] "Naive Bayes classification," [Online]. Available: https://www.ibm.com/docs/en/db2/9.7?topic=classification-naive-bayes.

[5] S. Maitra, "Prediction & Calibration Techniques to Optimize Performance of Machine Learning Models," 30 September 2019. [Online]. Available: https://towardsdatascience.com/calibration-techniques-of-machine-learning-models-d4f1a9c7a9cf. [Accessed 10 September 2021].