Dean&Francis

ISSN 2959-6157

Prediction of Diabetes Based on Machine Learning Algorithm

Zijian Zhou

Birmingham Joint Institute, Jinan University, Guangzhou, China

andy3619@stu2022.jnu.edu.cn

Abstract:

Diabetes is a well-known chronic disease that includes a range of metabolic disorders characterized by persistently elevated blood sugar levels over an extended period of time. Early and precise prediction of diabetes is essential to reduce risk factors and minimize potential complications associated with the disease. However, there are significant challenges in creating reliable predictive models due to factors such as limited labeling data, the presence of outliers, and the absence of information in diabetes-related datasets. To address these barriers, this paper proposes a comprehensive framework aimed at improving diabetes prediction through data preprocessing and machine learning techniques. The framework combines methods for dealing with missing values, data standardization, and feature visualization to extract meaningful insights. In addition, various machine learning classifiers - including support vector machine (SVM), decision tree, logistic regression, and naive Baye - are implemented to improve prediction accuracy and support early diagnosis of diabetes. Among these models, SVM shows better comprehensive performance.

Keywords: Diabetes; Missing value; Feature visualization; Machine learning.

1. Introduction

Diabetes is a chronic illness that raises blood sugar levels because the body either cannot use the insulin it does generate properly or does not produce enough of it. Type 1 (autoimmune-related) and Type 2 (often associated with lifestyle variables) comprise the majority. However, when the body is unable to produce sufficient insulin or properly utilize it, the glucose stays in the blood instead of reaching cells. Diabetes mellitus (DM) is caused by improper absorption of nutrients, resulting in abnormal blood sugar levels. Prevention strategies such as maintaining a balanced diet and adopting a healthier lifestyle are crucial, as poor nutrition or obesity can be major contributing factors to diabetes. These measures also help control blood pressure and reduce the likelihood of other health complications. This paper introduces a framework for early detection of diabetes based on machine learning.

2. Literature References

Many methods for predicting diabetes have been developed and published recently. Reference [1] introduces a machine learning (ML) framework, to which the authors apply many applications, including Linear Discriminant Analysis [2], Quadratic Discriminant Analysis [3], Naive Bayes [4], Gauss Process Classification [5], Support Vector Machine [6], Artificial Neural Network [7] and so on. Approaches like AdaBoost [8], Decision Tree [9], and Random Forest [10] employ different cross-validation approaches and dimensionality reduction techniques. They also carried out a number of experiments to improve the ML model's performance, reaching a maximum AUC of 0.930. These experiments included processing missing data and removing outliers. Three distinct classifiers—dt, SVM, and NB—were employed by the researchers to estimate the likelihood of diabetes; NB performed best, with an AUC of 0.819. For the categorization of diabetes, integrated methods such AdaBoost and bagging based on J48 Dt-based learners were examined. According to their findings, AdaBoost performs better than both independent and bagged J48-DT models.

This paper presents a new method for predicting diabetes using PIMA Indian diabetes data set. At the heart of the pipeline are pre-processing steps, including outlier removal, missing data entry, data set standardization, and selection of relevant features.

3. Methodology

3.1 Preprocessing the data and visualizing the features

In order to improve the performance and effect of the diabetes prediction model, it is necessary to pre-process the original data, modify or delete the data that is not suitable for the model or inaccurate data, and finally make the pre-processed data conform to the model requirements for the model. The data preprocessing method is as follows:

l Missing value processing is an important part of data preprocessing. Ensuring the integrity, accuracy and reliability of the data is critical. In Fig. 1, the percentage of missing values of each parameter is different.

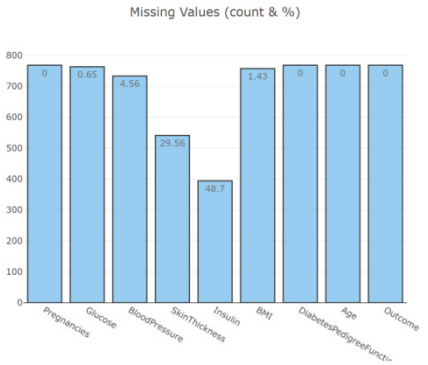


Fig. 1 The bar chart of missing values

l In the example of calculating the missing value of insulin, as Table 1 shows that the median of each parameter is calculated programmatically, and then each missing value is replaced with the median and the related image is drawn. Glucose, skin thickness, blood pressure and body mass index are processed in the same sequence.

If the number is odd, the median is:

$$M = X_{\frac{n+1}{2}} \tag{1}$$

If the number is even, the median is:

$$M = \frac{1}{2}(X_{\frac{n}{2}} + X_{\frac{n}{2}+1}) \tag{2}$$

Where M is the median, X is the sequence, and n is the quantity of sequence

Table 1. The median of Insulin

	Outcome	Insulin
0	0	102.5
1	1	169.5

3.2 Drawing the correlation matrix

The role of Correlation Matrix is to visualize the relationships between the variables in the data set. The correlation matrix shows the correlation between each feature and the other features, with a correlation value between -1 and 1. Values near 1 suggest a strong positive correlation, meaning both features increase together, while values near -1 suggest a strong negative correlation, where one feature rises as the other falls. A value around 0 indicates little to no correlation between the features. This step can help identify which features are strongly correlated with each other, which can be used as a reference for subsequent processing, such as whether redundant features need to be removed, combined features, or interactions between fea-

ISSN 2959-6157

tures need to be explored to optimize the machine learning model.

The Pearson correlation coefficient is calculated as follows:

$$r_{X} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \quad (3)$$

3.3 Machine learning methods

The proposed diabetes classification and prediction system utilizes various machine learning algorithms. Logistic regression, SVM, naive Bayes and decision tree model were used to classify diabetes. Among them, SVM has been specifically fine-tuned due to its strong performance in healthcare applications, particularly in predicting diabetes.

It is appropriate to use logistic regression when the dependent variable is binary. A comparative analysis was performed to assess the effectiveness of these techniques in classifying individuals into different categories of diabetes. In addition, it is often used in predictive analysis to help clarify the relationship between the dependent variable and one or more explanatory variables, as shown in equation (4). By using the sigmoid function as an assumption, the goal is to minimize the cost function, resulting in a class 1 or class 2 classification.

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \quad (4)$$

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^i \log h_\theta(x^i) + (1-y^i)\log(1-h_\theta(x^i))] \quad (5)$$

The Naive Bayes model is particularly advantageous for diabetes prediction due to its simplicity, efficiency, and strong performance in handling high-dimensional data. Its probabilistic approach assumes feature independence, which, while a simplification, allows it to effectively classify instances even when features are not highly correlated. This characteristic is especially useful in medical datasets where features may be numerous but not always intricately related. Additionally, Naive Bayes requires relatively less computational power compared to more complex models, making it well-suited for quick predictions and real-time applications. Its capacity to effectively manage both numerical and categorical data boosts its usefulness across various datasets related to diabetes prediction.

Because SVM is resilient in classification tasks and can handle complex, high-dimensional information, it is very useful for diabetes prediction. Its role is to determine the ideal hyperplane, dividing as many classes as possible, which improves the model's ability to adapt to new data. This capacity is critical in medical settings where complex and non-linear feature interactions may exist, such as diabetes prediction. By using kernel functions, SVM provides even more versatility by allowing the translation of input characteristics into higher-dimensional spaces to capture complex patterns. Furthermore, SVM is a reliable option for precise diabetes classification since it is less prone to overfit, particularly in high-dimensional settings.

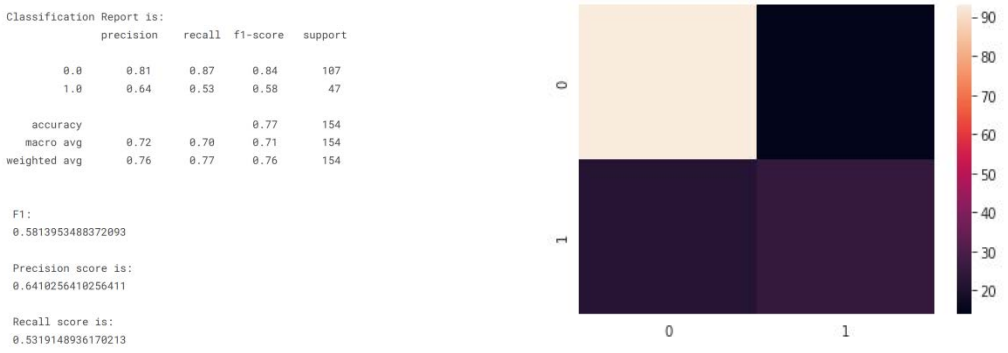
Decision trees are very useful for diabetes prediction because of their interpretability and efficient handling of both categorical and numerical data. Based on feature values, the model divides the data into subsets and builds a structure resembling a tree, with each node denoting a decision rule and each branch representing an outcome of the rule. This well-defined framework facilitates comprehension of decision-making processes, which is useful when elucidating forecasts in medical contexts. In order to effectively describe the complex aspects related with diabetes, decision trees must also be able to capture non-linear correlations between features and automatically handle feature interactions.

4. Results

4.1 Train-Test Data Split

The research object in this work is the Pima Indian diabetes data set.It is usual practice to divide a dataset into 80% training and 20% testing, for several reasons. The model can learn from a huge amount of data thanks to the training set, which aids in identifying underlying patterns and characteristics. Using the testing set helps to ensure that the model can prevent overfitting and generalize beyond the training set by evaluating its performance on fresh data. An independent measure of the effectiveness of the model is maintained by separating 20% of the data for testing purposes. This split strikes a compromise between having an adequate amount of data for testing and training, and it may be improved even further by using methods like cross-validation to evaluate model performance and adjust hyperparameters.

4.2 Naive Bayes

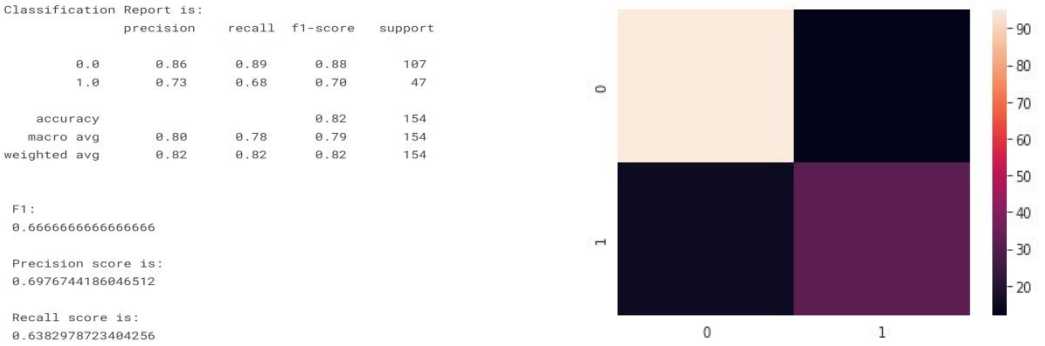


(a) (b)

Fig. 2 Result of Naive Bayes: (a)classification report; (b)confusion matrix

Figure 2 reveals a model with different levels of accuracy for the two classes. Class 0.0 exhibits a higher accuracy with an 0.81 accuracy rate and an 0.87 retrieval rate, meaning the model rarely errs in identifying this class. In contrast, class 1.0 has a lower accuracy rate of 0.64 and a retrieval rate of 0.53, indicating a higher chance of misclassification. The composite F1-measure for class 1.0 is 0.58, striking a balance between precision and sensitivity.77% of the samples are correctly classified overall, according to the 0.77 correct classification rate. The weighted metrics, which take into account the class distribution, exhibit a performance bias toward the more common class, while the macro-average metrics average out to a respectable performance level. With an accuracy rate of 0.64 and an F1-measure of 0.58 for the minority class, it appears that improving the model's handling of the class distribution could increase its efficacy.

4.3 Support Vector Machine



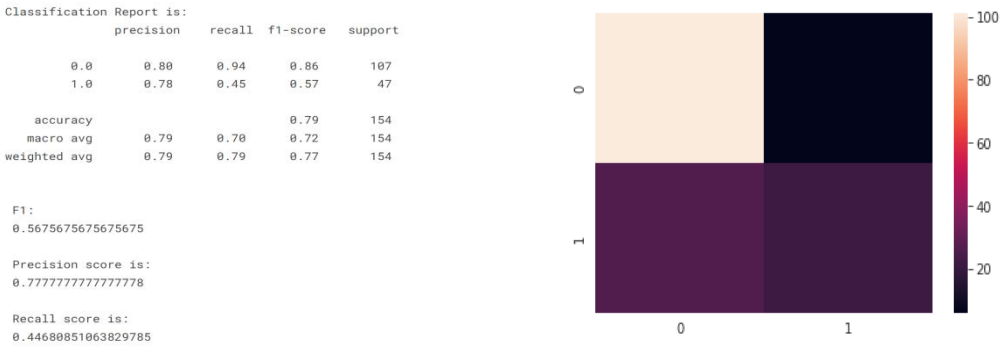
(a) (b)

Fig. 3 Result of SVM: (a)classification report; (b)confusion matrix

Figure 3 illustrates the improved model performance over the previous iteration, with an increase in overall accuracy to 82%. With a precision rate of 86%, a detection rate of 89%, and a balancing score of 0.88 for class 0.0, the model has strong performance, suggesting that it is very accurate and reliable at class identification. Class 1.0's performance was also markedly enhanced, exhibiting a true positive rate of 73%, a detection rate of 68%, and a balance score of 0.70. Although the model continues to perform better in class 0.0, the differences between classes have diminished, suggesting that recent tweaks have had a positive impact on the model's ability to recognize the less frequent class 1.0. The weighted average score further validates the overall improvement of the model, reflecting a more balanced performance between the two classes.

ISSN 2959-6157

4.4 Decision Tree



(a) (b)

Fig. 4 Result of Decision Tree :(a)classification report (b)confusion matrix

The model performs better in Figure 4 than it did in the prior iteration, with an overall accuracy increase to 79%. With a true positive rate of 80%, a detection rate of 94%, and a balanced score of 0.86 for class 0.0, the model exhibits remarkable performance, proving its excellent accuracy and reliability in class identification. With a true positive rate of 78%, a detection rate of 45%, and a balanced score of 0.57, class 1.0 has also shown a discernible improvement. The disparity between the classes has shrunk, even if the model still performs better for class 0.0; this suggests that the recent modifications have improved the model's capacity to identify the less common class 1.0.

4.5 Logistic Regression

(a) (b)

Fig. 5 Result of Logistic Regression :(a)classification report (b)confusion matrix

The model's performance in a binary classification test is shown in Figure 5, where it achieved an overall accuracy of 79%. With a true precision of 83%, a detection rate of 89%, and a balanced score of 0.86 for class 0.0, the model performs well and exhibits its high dependability in class identification. In contrast, class 1.0 exhibits a less robust performance from the model, with a balanced score of 0.63, a true positive rate of 69%, and a detection rate of 57%. This suggests that the model has trouble correctly recognizing the less prevalent class 1.0. These results imply that more optimization is required, especially to improve the model's capacity to identify class 1.0, which might be done by adjusting the model's parameters.

4.6 Comparison of model results

Table 2. Performance of different models

Model	Precision score	Recall score	F1-score
Naive Bayes	0.64	0.53	0.58
Support Vector Machine	0.70	0.64	0.67
Decision Tree	0.78	0.45	0.57
Logistic Regression	0.70	0.57	0.63

In Table 2, it is obvious that in terms of model accuracy, decision tree performs the best, while it has a poor recall score. SVM has higher recall rate and F1 score than the other three models, indicating that SVM has high stability and accuracy. Overall, SVM performs well.

5. Conclusion

In this paper, diabetes predictions were made using an integrated model developed from the PIMA Native American Diabetes dataset, and preprocessing was critical for accurate and reliable results. The proposed preprocessing method focuses on improving data set quality by processing missing values and generating correlation matrix. These pre-processing steps enhance the distribution's peakedness and asymmetry in the dataset.

While various models have been used to predict diabetes, synthesizing these models effectively has been a major challenge for researchers. Therefore, it is crucial to identify models with strong overall performance. In this study, a predictive classification model for diabetes based on machine learning was proposed. Several models were trained and tested on sample data, and the support vector machine with the highest overall performance was selected because of its superior predictive ability.

References

[1] Maniruzzaman M, Rahman M J, Al-MehediHasan M, et al. Accurate diabetes risk stratification using machine learning: role of missing value and outliers. Journal of medical systems, 2018, 42: 1-17.

[2]McLachlan G J. Discriminant analysis and statistical pattern recognition. John Wiley & Sons, 2005.

[3]Cover T M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE transactions on electronic computers, 1965 (3): 326-334.

[4]Webb G I, Boughton J R, Wang Z. Not so naive Bayes: aggregating one-dependence estimators. Machine learning, 2005, 58: 5-24.

[5]Brahim-Belhouari S, Bermak A. Gaussian process for nonstationary time series prediction. Computational Statistics & Data Analysis, 2004, 47(4): 705-712.

[6]Cortes C. Support-Vector Networks. Machine Learning, 1995.

[7]Reinhardt A, Hubbard T. Using neural networks for prediction of the subcellular location of proteins. Nucleic acids research, 1998, 26(9): 2230-2236.

[8]Kégl B. The return of AdaBoost. MH: multi-class Hamming trees. arxiv preprint arxiv:1312.6086, 2013.

[9]Jenhani I, Amor N B, Elouedi Z. Decision trees as possibilistic classifiers. International journal of approximate reasoning, 2008, 48(3): 784-807.

[10]Breiman L. Random forests. Machine learning, 2001, 45: 5-32