Logistic Regression for Stroke Prediction: An Evaluation of its Accuracy and Validity

Lechi Wang *

Department of Software Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China

* Corresponding author email: b20032006@njupt.edu.cn

Abstract. A stroke, also known as a brain attack, is a serious medical condition that occurs when the blood supply to the brain is disrupted. It is a leading cause of death globally, accounting for about 11% of all deaths. In this paper, the author opts to use logistic regression for predicting the stroke. The paper starts with introducing the methods used to preprocess the raw dataset, including data cleaning, label encoding, oversampling, splitting the dataset and finally feature scaling. Then it goes to the modeling section. In this section, the author introduces the process of constructing a logistic regression model in detail. The first step is building the sigmoid function, a foundation of the model. Then the cost function is built to measure the difference between the model's predicted values and the true values. After that, the author constructs the gradient-computing function to determine the rate and direction at each iteration when implementing gradient descent. Based on this function, the gradient descent function is finally built, and predictions are made according to the outcome after putting testing variables into the model. By analyzing the results, the author also compares the different performances of the model between the one without regularization and the one equipped with regularization and then draw a conclusion that using regularization can help improve the performance of the model. At the end of the study, the author gets a rather satisfying result with a prediction accuracy of over 95% in the collected dataset.

Keywords: Stroke Prediction; Logistic Regression; Gradient Descent; Regularization.

1. Introduction

A stroke, sometimes called a brain attack, occurs when something blocks the blood supply to part of the brain or when a blood vessel in the brain bursts. In either case, parts of the brain become damaged or die. A stroke can cause lasting brain damage, long-term disability, or even death. Furthermore, stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths. According to the research of Centers for Disease Control and Prevention (CDC), every 40 seconds, someone in the United States has a stroke; every 3.5 minutes, someone dies of the stroke [1]. Therefore, lucubrating this kind of disease to gain a better understanding for it is rather indispensable. There are many approaches to gain insights into the stroke. In this case, making predictions on stroke is one of the major means to detect and prevent this kind of disease.

Long before the computer era, doctors had been making predictions on various kinds of disease, including stroke [2]. At that time, the evidence for making predictions was limited to their career experience and patients' symptoms related to stroke. In other words, they lacked scientific indicators for accurate predictions. With the advent of machine learning and artificial intelligence, researchers start to implement algorithms on the model building to make predictions in a much more scientific way in many tasks [3-7]. However, most of them typically only focuses on building multiple models and comparing their performance of predicting stroke. What they often pay less attention to or even ignore is the data visualization and preprocessing part. Moreover, due to the huge amounts of algorithms implemented, they do not thoroughly explain the principle of each algorithm; instead, they just simply call functions to achieve their purpose. For example, one article focuses on the analysis of machine learning approaches in stroke prediction [8]. The authors of this article compare the performance of ten machine learning classifiers including e.g., LR, SGD, DTC, Linear regression, AdaBoost, Gaussian, QDA, MLP, KNeighbors, GBC and, XGB. and weighted voting classifier. At the end of the article, the authors draw a conclusion that weighted voting should be considered for

the prediction of stroke. Nevertheless, the process of data exploratory analysis and data preprocessing is briefly clarified. In addition, all the machine learning algorithms in this article are achieved by utilizing Python and Scikit-learn libraries. As a result, though revealing lots of profound insights, the article fails to present the whole process of a prediction as well as a clear explanation of the principle of a certain algorithm and the reasons why the authors choose that algorithm for implementation.

In this regard, this paper adopts a more comprehensive approach to do the research. The first section is about data cleaning, where this paper focuses on neatening the raw dataset by means of dealing with missing values and outliers. To get a better understanding of the correlation between stroke and the potential factors, this paper designs the section of data visualization. Furthermore, since directly using the raw dataset for modeling without any modification is not likely to produce a satisfying outcome, this study also spends effort in preprocessing the data, including handling with oversampling and doing feature scaling. The last section, also the core section, is modeling. In this section, this study implemented one machine learning algorithm: the logistic regression algorithm. Instead of using the one from Scikit-learn, this study constructed the algorithm from scratch and made adjustment based on the model evaluation which comes afterwards. In the end, the author gets a prediction model with both the accuracy score and F1 score over 95%.

2. Method

2.1 Dataset Description and Preprocessing

In this paper, the dataset is acquired from the website of Kaggle [9]. This dataset is used to predict whether a patient is likely to get stroke based on the input parameters. Each row in the data provides relevant information about the patient. The data contains 5110 rows with 12 features. This dataset is applied to build a classification model. Apparently, “stroke” is the dependent variable, which is a feature with binary classes. All features included in this dataset can be found in Table 1.

Table 1. All features and corresponding description in the dataset

Feature name	Description
Id	unique identifier
gender	"Male", "Female" or "Other"
age	age of the patient
hypertension	0 if the patient doesn't have hypertension, 1 if the patient has hypertension
heart_disease	0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
ever_married	"No" or "Yes"
work_type	"children","Govt_jov", "Never_worked", "Private" or "Self-employed"
Residence_type	"Rural" or "Urban"
avg_glucose_level	average glucose level in blood
bmi	body mass index
smoking_status	"formerly smoked", "never smoked", "smokes" or "Unknown"
stroke	1 if the patient had a stroke or 0 if not

The paper divides the process of preprocessing data into four steps, namely Data cleaning, Label encoding, Oversampling and Splitting the dataset & Feature scaling.

2.1.1 Data Cleaning

In this analysis, the 'id' column is removed as it has no relevance to the stroke prediction. Additionally, this paper drops the single row where the 'gender' value is recorded as 'Other'. There are no duplicate values in the dataset. However, there are nearly 200 rows with missing values in the 'bmi' column, which are replaced with the mean value of the other rows. There are 1544 rows with unknown 'smoking status', and the largest proportion of these rows are samples under the age of 18.

Instead of discarding these rows, the author opts to change the 'smoking_status' of samples under 18 from 'unknown' to 'never_smoked', as it is unlikely that these individuals have smoked.

2.1.2 Label Encoding

In order to more accurately assess the influence of each independent variable on the dependent variable, dummy variables are created for the initial categorical variables (e.g., gender, work_type). By using dummy variables, this study can more directly assess the influence of each category of the original categorical variable on the dependent variable."

2.1.3 Oversampling

One issue the author encountered while preparing the data for analysis was a significant imbalance between the number of samples labeled as 'stroke=1' and 'stroke=0'. Specifically, there are far fewer 'stroke=1' samples compared to 'stroke=0' samples. To address this issue, this paper uses the SMOTE function from the imblearn library to oversample the 'stroke=1' samples. Oversampling involves generating synthetic samples that are similar to the existing 'stroke=1' samples, with the goal of increasing the overall number of 'stroke=1' samples to better balance the dataset. By using SMOTE, this study is able to more accurately reflect the true prevalence of stroke in the population and improve the performance of our predictive model.

2.1.4 Splitting the Dataset

Before building the logistic regression model, it is necessary to split the dataset into a training set and a testing set. The author used a ratio of 70% training data and 30% testing data. The training set is used to fit the model, while the testing set is used to evaluate the model's performance.

2.1.5 Feature Scaling

The last step in preprocessing the data is feature scaling. Feature scaling involves standardizing the range of values for each feature so that they are on a similar scale. By scaling the features, the model will not be unduly influenced by the scale of the data. Additionally, scaling the features can help to speed up the rate of convergence when applying gradient descent, which will be introduced afterwards.

2.2 Logistic Regression

Logistic regression is a type of regression machine learning algorithms being deployed to solve classification problems [10]. Problems having binary outcomes, such as Yes/No, 0/1, True/False, are the ones being called classification problems.

In this paper, sigmoid function is the first to be built. Then, it is implemented to the upcoming process including calculating cost function and computing gradient descent. What comes next is applying gradient descent. Finally, the model is finished being built and predictions are made based on it. The complete workflow can be found in Figure 1.

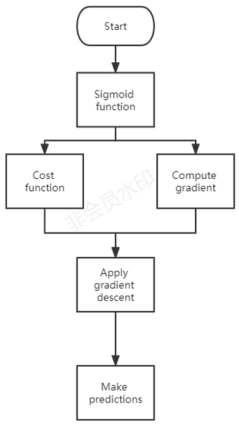


Fig 1. The workflow of logistic regression

2.2.1 Sigmoid Function

$$S(x) = \frac{1}{1+e^{-x}} \tag{1}$$

The main purpose of building sigmoid function is to map variables between 0 and 1. In this paper, the sigmoid function contains one parameter which represents the input value. The function returns the corresponding result after calculation.This specific calculations performed by the sigmoid function are designed to produce this non-linear relationship between the input and output values.

2.2.2 Cost Function

$$J(\theta) = \frac{1}{\mathrm{m}}[\sum_{i=1}^{m} - y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))] \tag{2}$$

The primary purpose of cost function is to measure the difference between the model's predicted values (h(θ)) and the true values of the target variable. By minimizing the cost function, the model's performance can be improved and the error between the predicted and true values can be reduced.

In this paper, l2 regularization is used when defining the cost function. It introduces a penalty term to the cost function that penalizes models with large weights, which helps to reduce the complexity of the model and improve its generalization ability. By adding this penalty term to the cost function, the model will not be overfitted to the training data and will be better able to make accurate predictions on new, unseen data.

2.2.3 Computer Gradient

$$\begin{aligned} \frac{\partial J(w,b)}{\partial w_j} &= \frac{1}{m}\sum_{i=0}^{m-1}(f_{w,b}(x^{(i)}) - y^{(i)})x_j{}^{(i)} \\ \frac{\partial J(w,b)}{\partial b} &= \frac{1}{m}\sum_{i=0}^{m-1}(f_{w,b}(x^{(i)} - y^{(i)}) \end{aligned} \tag{3}$$

The main purpose of computing gradient is to determine the partial derivative of w and b at each iteration of gradient descent. The partial derivative represents the rate of change of the cost function with respect to a particular parameter, and it also helps to determine the direction in which the model should update the parameter to minimize the loss function. By iteratively updating the parameters in the direction of the gradient, gradient descent is able to quickly and efficiently find the values that minimize the loss function.

2.2.4 Apply Gradient Descent

$$\begin{aligned} &\text{repeat until convergence}\{ \\ &w_j = w_j - \alpha \frac{\partial J(w,b)}{\partial w_j} \quad \text{for j:=0..n-1} \\ &\mathrm{b} = b - \alpha \frac{\partial J(w,b)}{\partial b} \\ &\} \end{aligned} \tag{4}$$

Performing the gradient descent algorithm given the learning rate alpha, the number of iterations num_iters to find the best w and b which minimize the cost value.

To apply gradient descent, it is necessary to iterate over this pair of formulas multiple times until the model reaches a minimum of the loss function (i.e., the point where the gradient is zero). At this point, the model has learned the optimal values for the parameters and can be used to make predictions on new data.

2.2.5 Make Predictions

After training the model, the parameters (i.e. w and b) that were optimized during the training process are saved. These parameters represent the weights and biases of the model, which are used to make predictions on new data.

To predict the category of an input sample, the model applies the learned parameters to the input and produces a prediction value. This prediction value is then compared to a threshold, which is typically set at 0.5. If the prediction value is greater than or equal to the threshold, the model predicts the sample belongs to category "1". If the prediction value is less than the threshold, the model predicts the sample belongs to the other category "0".

3. Result and Discussion

The model's training accuracy and the corresponding F1 score can achieve 95.61% and 95.51% before regularization. In addition, Figure 2 and Figure 3 present the confusion matrix and ROC of the prediction results.

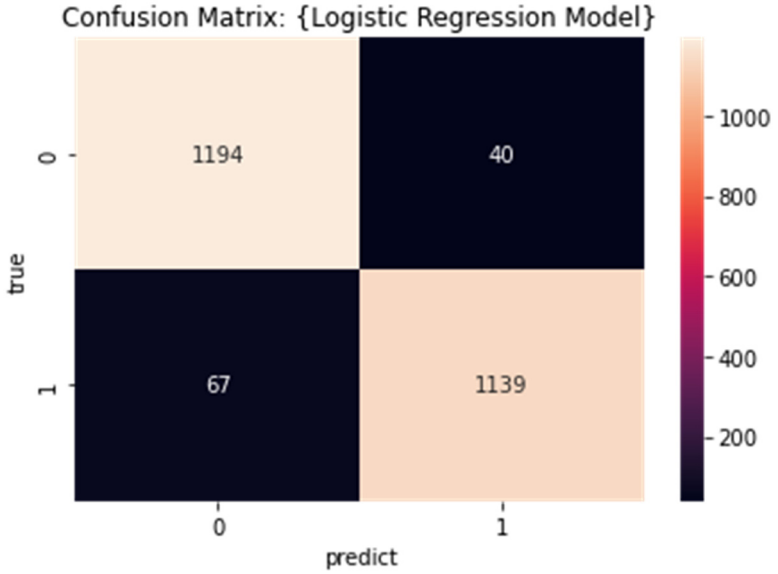


Fig 2. Confusion Matrix before regularization

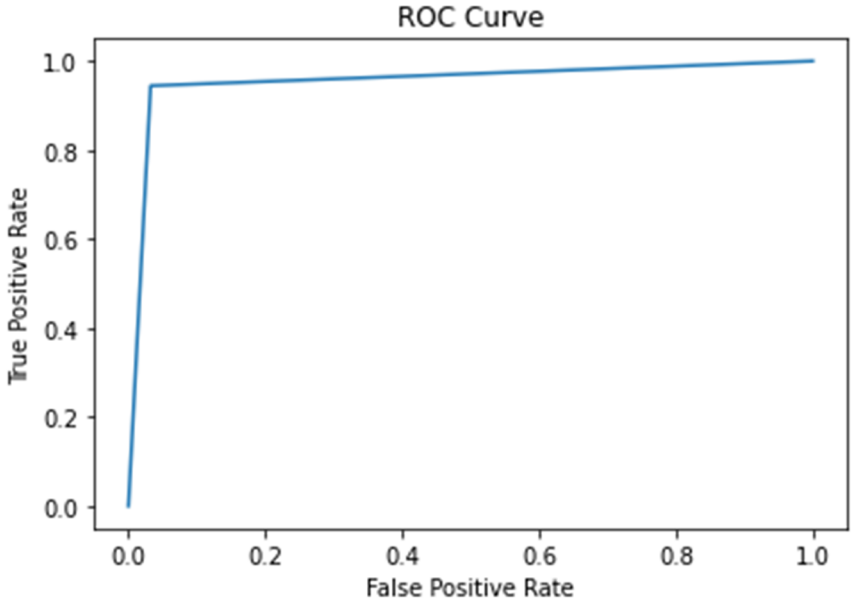


Fig 3. ROC curve before regularization

From the experimental results, it can be concluded that the model shows high accuracy and F1 score on the training set. The confusion matrix shows most of the data are correctly predicted, though a little of them are not. Also, the ROC curve presents a high TPR and a low FPR with high discriminative power. These indicate that the model fits the data well and has high predictive power. However, considering the risk of overfitting, it is necessary to implement regularization methods to prevent it.

The model's training accuracy and the corresponding F1 score can achieve 95.98% and 95.83% after regularization. In addition, Figure 4 and Figure 5 present the confusion matrix and ROC of the prediction results.

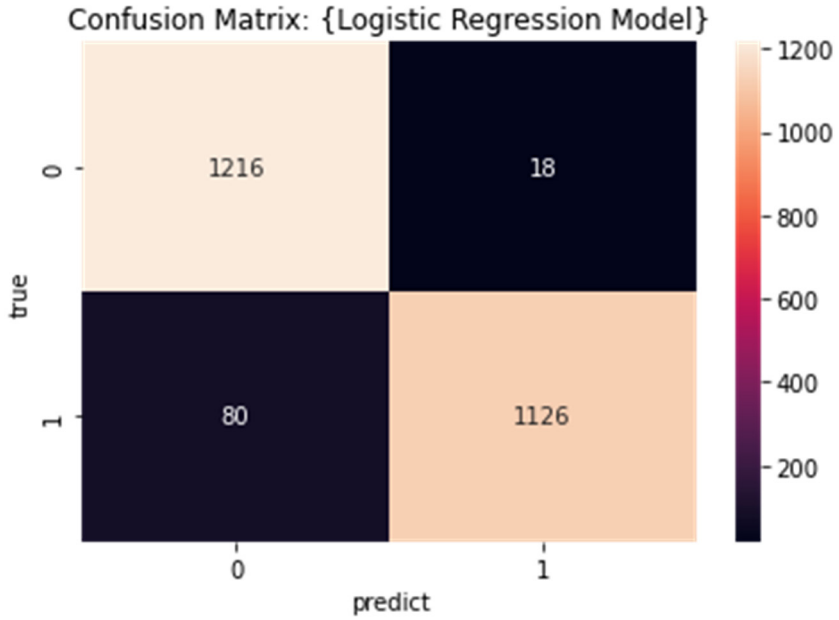


Fig 4. Confusion Matrix after regularization

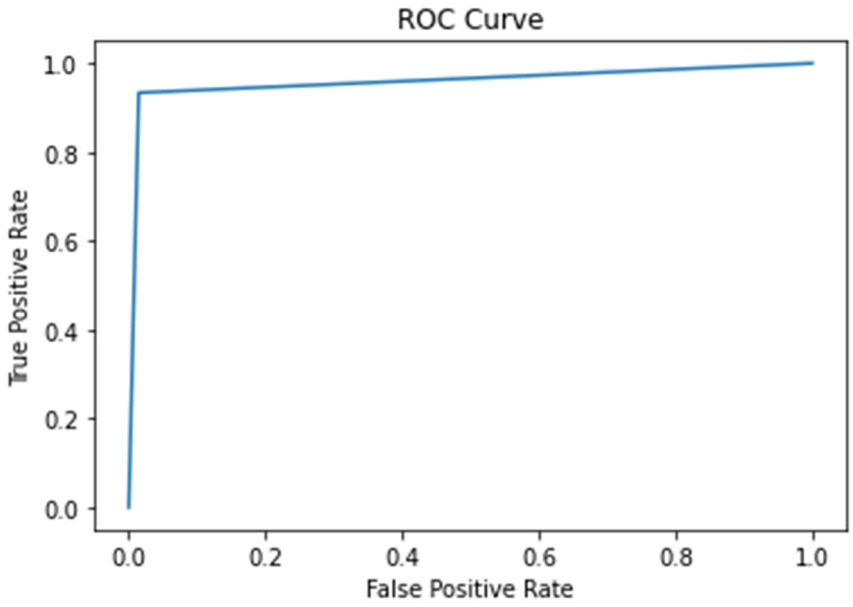


Fig 5. ROC curve after regularization

It shows that the accuracy figure and F1 figure do raise up a bit compared to the one without regularization. However, the confusion matrix as well as the ROC curve does not vary much after the implementation.

There are two reasons that possibly account for this issue: One reason could be that regularization helps to prevent overfitting, which can improve the overall performance of the model. However, the confusion matrix and ROC curve are measures of how well the model can classify individual instances, and so small changes in these metrics may not be noticeable if the overall performance of the model has improved.

Another reason could be that the improvement in the accuracy and F1 score is not significant enough to have a noticeable effect on the confusion matrix or ROC curve. In this case, even though the model may have improved slightly with regularization, the difference may not be large enough to be reflected in the confusion matrix or ROC curve.

4. Conclusion

In conclusion, this study has demonstrated the effectiveness of using a logistic regression approach in predicting the likelihood of a stroke. The results in this study show that this method achieves a high level of accuracy, with a prediction accuracy of over 95% in the collected dataset. These findings suggest that this approach may be useful in identifying individuals at high risk for stroke, and could potentially be used in clinical settings to guide preventative measures. /Nevertheless, further research is needed to confirm these results and to explore the potential applications of this method in a broader range of populations. In addition, some advanced algorithms e.g. Random forest and neural network will be also considered to further improve the prediction performance.

References

[1] CDC. Stroke. https://www.cdc.gov/stroke/facts.html, 2022.

[2] Meretoja, Atte, et al. Stroke doctors: who are we? A world stroke organization survey. International Journal of Stroke 12.8, 2017, 858-868.

[3] Shinde, Pramila P., and Seema Shah. A review of machine learning and deep learning applications. 2018 Fourth international conference on computing communication control and automation (ICCUBEA). IEEE, 2018.

[4] Libbrecht, Maxwell W., and William Stafford Noble. Machine learning applications in genetics and genomics. Nature Reviews Genetics 16.6, 2015, 321-332.

[5] Q, Yu, et al. Improved denoising autoencoder for maritime image denoising and semantic segmentation of USV. China Communications 17.3, 2020, 46-57.

[6] Kourou, Konstantina, et al. Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal 13, 2015, 8-17.

[7] Kayalibay, Baris, Grady Jensen, and Patrick van der Smagt. CNN-based segmentation of medical imaging data. arXiv preprint arXiv:1701.03056, 2017.

[8] Emon, Minhaz Uddin, et al. Performance analysis of machine learning approaches in stroke prediction. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2020.

[9] Kaggle. Stroke prediction dataset. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset, 2020.

[10] Wright, Raymond E. Logistic regression, 1995.