Stroke Risk Prediction Based on Machine Learning Algorithms

Zijun Chen*

Department of School of Intelligent Systems Science and Engineering, JINAN University, Zhuhai, 519070, China

* Corresponding Author Email: cczzj@stu2020.jnu.edu.cn

Abstract. Stroke is one of the main causes of long-term disability and death around the world. For its significant impact, stroke is also defined as a medical emergency, where immediate treatment is vital to save patients' lives. For these reasons, an efficient prediction plays an important role in stroke prevention and cure. In this paper, attempting to implement a model predicting stroke efficiently, logistic regression and random forest algorithms are adopted, as well as a stroke dataset. They are trained and make predictions with the preprocessed dataset independently. Multiple evaluation indicators are employed to evaluate the two models' results. Comparisons between their performances and the reasons for their discrepancies are both introduced, based on which the more suitable one is chosen as the final model. Models' bias and variance and how they influence the results are discussed as well. In addition, some helpful propositions and approaches to improve the model's performance will also be introduced.

Keywords: Stroke prediction, machine learning, logistic regression, random forest.

1. Introduction

Stroke is a serious disease that usually happens when the supply of blood and oxygen to human brain is cut off, causing most brain cells to die in a short time [1, 2]. It is one of the principal causes of long-term disability and death around the world. In 2016, about 13.7 million people got a stroke, and 5.5 million people died from a stroke [3]. A stroke is defined as a medical emergency, where urgent medical treatment is essential to save patients' lives, that's why stroke prediction is so essential. If it could be diagnosied whether a person may suffer from a stroke based on some factors like blood pressure or smoking status, medical treatment could be conducted at an early age, which contributes significantly to reducing stroke risk and harm [4].

Nowadays, machine learning methods are applied to more and more medical areas [5]. Compared with traditional methods, machine learning is of higher efficiency and accuracy, thus playing an irreplaceable role in disease prediction and analytics [6, 7]. Before being put into use, machine learning methods or algorithms are trained under a large number of data, with which the algorithms can find the inner connections between factors and targets and then update themselves. After the training process, it is required to input the patient's record into the pre-trained model, the target or output will be given in a short time with an accuracy of up to 90%.

This study's purpose is to predict the occurrence of stroke, put another word, whether a patient will develop a stroke with the assistance of the machine learning method [8, 9]. A dataset containing patients' information is adopted. After preprocessing the dataset, two machine learning algorithms, logistic regression, and random forest are employed in two models' training process respectively. In general, these two algorithms are both employed to solve classification problems. However, their inner principles differentiate to a great extent, thus influencing prediction accuracy. For these reasons, comparisons and analyses between the two models are introduced as well, based on which the final model is chosen.

2. Method

It contains three parts. The first part is the information about the dataset, such as its size, features, and labels. In addition, the preprocessing procedure is introduced. The second part is the

implementation of Logistic Regression. Thirdly, the implementation of Random Forest is included in the comparison.

2.1. Dataset

This stroke dataset [10] is employed in the prediction whether a patient will develop a stroke according to the features, such as gender, age, and hypertension. Each patient's information are provided by a row. Each patient has 11 features, and one labels (stroke). 1 means the patient got a stroke while 0 means not. Representative samples are shown in Table 1.

Table 1. Representative example of the dataset

Id	**Gender**	**Age**	**Hypertension**	**Heart_disease**	**Ever_married**
9046	male	67	0	1	yes

Work_type	**Residence_type**	**Avg_glucose_level**	**Bmi**	**Smoking_status**	**Stroke**
private	urban	228.69	36.6	formerly smoked	1

2.2. Preprocessing

To improve the performance of the models, the data need to be processed before it is used to train the model. Firstly, delete useless features. Id is only a unique identifier of a patient. Unlike other features, it is little connected to the patient, influencing the performance in a way. Therefore, the ID of each patient should be delected. Secondly, handle null values. Some samples lack valid BMI values. These blanks are filled with the average of 'Bmi'. Thirdly, label the nonnumerical features. In the dataset, there are some features presented in a nonnumerical way, such as residence_type. These non-numerical values can't be understood by algorithms. Before using the dataset, it is required to convert it into a form that the model can recognize. Here label encoding method is introduced. Fourthly, split the data into features and labels. As it is supervised learning, features are the model's input while labels are the output. It will be useful to separate them into X and Y, representing features and labels respectively. After that, X is scaled with Z-score normalization. In the next step, a training set and a validation set are split from the dataset, the ratio between which is 4:1. In the last step, deal with the imbalanced problem. The proportion of patients who have stroke only represents 5%. To make models work better, improving label 1's proportion is necessary. Here SMOTE, an oversampling method is adopted. It adds to the number of labels 1 by sampling according to its inner algorithm.

2.3. Models

Since it is a binary-classification problem, logistic regression is employed. Besides, in order to better evaluate the model's performance, a random forest is employed in comparison.

2.3.1 Logistic Regression

Logistic regression is a linear regression model, belonging to supervised learning. It is mostly used in binary classification, though it can be generalized to deal with categorical classification as well. It is widely used for its simple implementation, faster calculation speed, and less computation cost.

After choosing logistic regression as the model, how it works is required to be understood, and what should be done. The following is the implementation procedure of logistic regression.
Firstly, the model of logistic regression is presented below(1)

$$z = \boldsymbol{w}^T \boldsymbol{x} + b \tag{1}$$

In formula (1), **w** is the weights of x. b is the bias. z is not the predicted value of the model. It needs to be converted into $\hat{y}$, the true predicted value through an activation function (2). In logistic regression, the sigmoid function is choosen as the activation function (3).

$$\hat{y} = \sigma(z) \tag{2}$$

$$\sigma(z) = \frac{1}{1+e^{z}} \tag{3}$$

However, it is not enough to only know about $\hat{y}$. Loss function needed as well, with which the model could be optimized efficiently. The cross-entropy loss (4) is used here, where $y^{(i)}$ is the label of sample (i), $\hat{y}^{(i)}$ is the value predicted by model. It gets some advantages compared with other loss functions. Figure 1 presents its image when y equals 1. It could be observed from the image that loss increased when z is gradually away from 0. On the contrary, loss equals to 0 approximately the time z is more than 1. It helps with gradient descent in the following steps.

$$\mathcal{L}_{CE} = -y^{(i)} \log \hat{y}^{(i)} - \left(1 - y^{(i)}\right) \log\left(1 - \hat{y}^{(i)}\right) \tag{4}$$

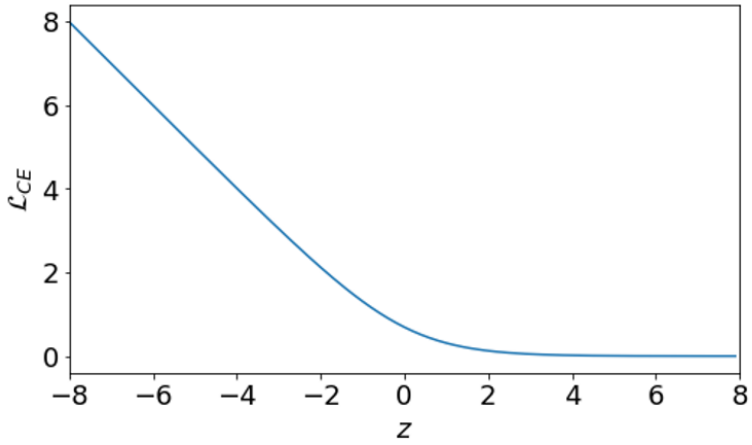


Fig. 1 Image of cross-entropy loss

Now the loss function is ready, then how to update the model with it? The origin gradient descent method is adopted. Firstly, the partial derivates of w and b should be calculated concerning loss (5) and (6), where w_i is the weight of the ith feature.

$$\frac{\partial \mathcal{L}_{CE}}{\partial w_i} = (\hat{y} - y) x_i \tag{5}$$

$$\frac{\partial \mathcal{L}_{CE}}{\partial b} = (\hat{y} - y) \tag{6}$$

Then calculate the sum of the loss of each sample together, and the gradient of w and b is achieved by (7), where $\boldsymbol{w}$ are the weights of **x**.

$$\nabla \mathcal{J}(\boldsymbol{w}, b) = \begin{bmatrix} \frac{\partial \mathcal{J}}{\partial \boldsymbol{w}} \\ \frac{\partial \mathcal{J}}{\partial b} \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathcal{L}_{CE}}{\partial \boldsymbol{w}} \\ \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathcal{L}_{CE}}{\partial b} \end{bmatrix} \tag{7}$$

Now the model can update itself with gradients. α here means learning rate, a hyperparameter in gradient descent algorithm. Its value determines the performance of the model. In general, the appropriate α can be got by trying.

$$w_i = w_i - \alpha \frac{\partial \mathcal{J}}{\partial w_i} \tag{8}$$

$$b = b - \alpha \frac{\partial \mathcal{J}}{\partial b} \tag{9}$$

In this work, after preprocessing, each sample has 10 features in total, from gender to smoking_status. Which means 10 weights w_i $(i = 0, 1, \ldots, 10)$ and a bias b should be initialized for original parameters. After that, input the dataset into the model, update parameters via the loss function, and the classification results are given by the model.

2.3.2 Random Forest

Random forest is a machine learning algorithm, belonging to supervised learning, which is widely used for classification and regression problems. Being a kind of ensemble learning, random forest works by constructing multiple decision trees, and gets the result through a majority vote in classification problems and calculating the mean value of predictions in the case of regression problems.

A random forest is a combination of multiple decision trees. And the training method for it is called bagging, which is also called bootstrap aggregation in decision trees' training procedure. Generally, bagging consists of the following steps:

Firstly, bootstrap samples as multiple training sets from the dataset. There are B trees in total, and B determines the number of the decision tree that are trained. It can be seen clearly form Figure 2 that bagging selects samples from the dataset with replacement as the training sets, repeatedly for B times. $\mathcal{D}$ is the dataset, $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ are multiple training sets

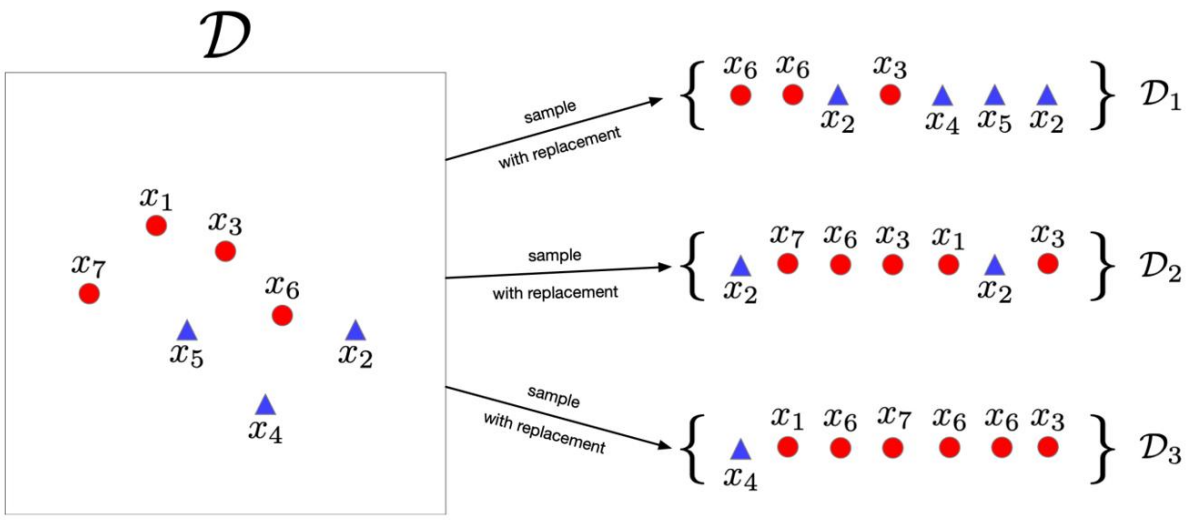


Fig. 2 Image of bootstrapping

Secondly, fit multiple decision trees to the training sets selected. After training, input the validation set and get the predictions for these samples. In the case of regression, predictions are made by calculating the average of all decision trees' predictions via formula (10), where *f* is the prediction for *x*, *x* is a sample in validation set. On the other hand, the majority vote method or average calculation is taken in the case of classification formulas (11) and (12), where z_i is the prediction of each decision tree

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x) \tag{10}$$

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x) \tag{11}$$

$$\mathbb{I}(z > 0.5) = \begin{cases} 1, z > 0.5 \\ 0, z \leq 0.5 \end{cases} \tag{12}$$

However, general bagging usually comes with some problems in the case of random forest. For example, if one or two features are strongly correlated with the target predictions, they are more likely to be chosen by decision trees during the training process. This leads to a high correlation between trees, affecting the model performance in a way.

To fix this, a modified bagging method is employed. Compared with the original one, it doesn't select all the features of the sample, but a random subset of the features instead, which is also called features bagging. It contributes to reducing the correlation between trees and thus promotes random forest performance.

2.3.3 Evaluation Matrix

To better evaluate the performance of the models, this study employed the evaluation indicators below as measurement.

Accuracy. Accuracy indicates the ratio between the number of correct predictions and the number of total predictions (13).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{13}$$

Precision and recall. Precision is the percentage of positive correct predictions among all true positive plus false positive predictions (14). The recall is the percentage of positive correct predictions among all true positive plus false neagtive samples (15).

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

$$Recall = \frac{TP}{TP+FN} \tag{15}$$

F_1-score. F_1-score can be seen as a weighted mean of precision and recall (16).

$$F_1 = \frac{2\times precision\times recall}{precision+recall} \tag{16}$$

Confusion matrix. Table 2 presents the detailed information of confusion matrix. s.

Table 2. Confusion matrix.

Total = TP + FN + FN + TN		Predicted labels	
		Positive	Negative
Actual labels	Positive	TP	FN
	Negative	FP	TN

3. Result

To better evaluate the performance of the algorithms, multiple evaluation indicators are employed, including accuracy, precision, recall, F_1-score, and confusion matrix. In addition, histogram about accuracy comparison is introduced as well.

The following Table 3 and Table 4 present the performance of two models in many aspects. The first raw presents evaluation indicators, precision, recall, and F_1-score are all included. The second and third raw gives the values of labels 0 and 1 on each indicator. The final raw reveals the accuracy of this model. Obviously, random forest performs better than logistic regression.

Table 3. Performance of Random Forest algorithm

RF	PRECISION	RECALL	$\boldsymbol{F_1}$-SCORE	SUPPORT
0	0.97	0.95	0.96	998
1	0.95	0.96	0.96	947
ACCURACY			0.96	1945

Table 4. Performance of Logistic Regression algorithm

LR	PRECISION	RECALL	$\boldsymbol{F_1}$-SCORE	SUPPORT
0	0.82	0.74	0.78	998
1	0.75	0.82	0.79	947
ACCURACY			0.78	1945

The following two figures Figure 3 and Figure 4 present two models' confusion matrix. Predicted labels and true labels are presented in x and y axises respectively. It visualizes models' performance from which the results of these two models could be compared in an intuitive way.

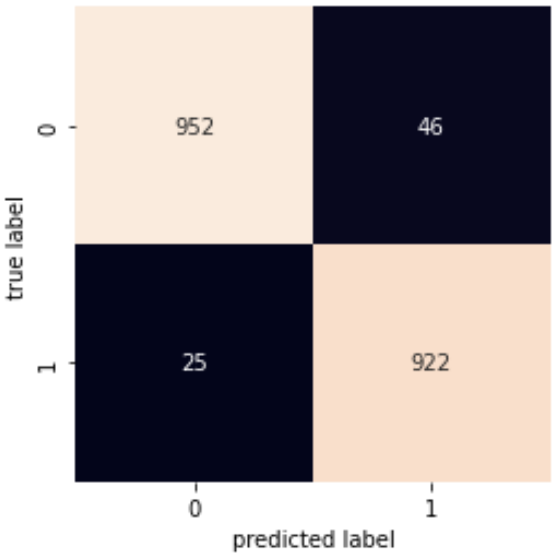


Fig. 3 Confusion matrix of Random Forest

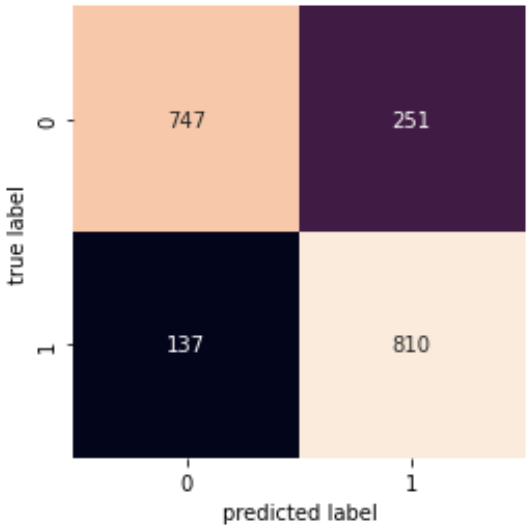


Fig. 4 Confusion matrix of Logistic Regression

The histogram in Figure 5 shows the accuracy comparison of two models. Obviously, random forest is nearly 20 percent higher than logistic regression.

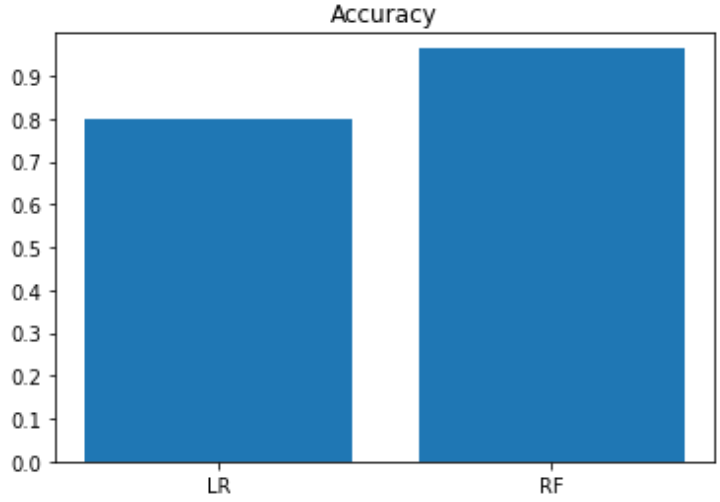


Fig. 5 Image of accuracy comparison between RF and LR

4. Discussion

In this part, two main questions will be discussed. Firstly, what cause the performance discrepancies between two models, inner or outer causes? Secondly, the accuracy of logistic regression model stands below 80%, and the performance of random forest seems still improvable. Are there any propositions or solutions can be applied to solve this problem?

4.1. Bias And Variance

Above all, model's bias and variance need to be discussed. Bias and variance are concepts used in supervised learning, belonging to reduceable error, which commonly occur when model is too simple or complex to fit the dataset. Bias indicates the amount that model's predictions differ from the targets, while variance indicates the amount of variability in the predictions. It could get a better understanding of it via the following Figure 6. In general, bias and variance are associated with model's complexity. If a model is not complex enough, thus it is not capable to find the relations between features and targets. It leads to high bias, which is also called underfitting. If a model is too complex, it tries to find out some unnecessary relations between features and targets, leading to high variance, called overfitting as well.

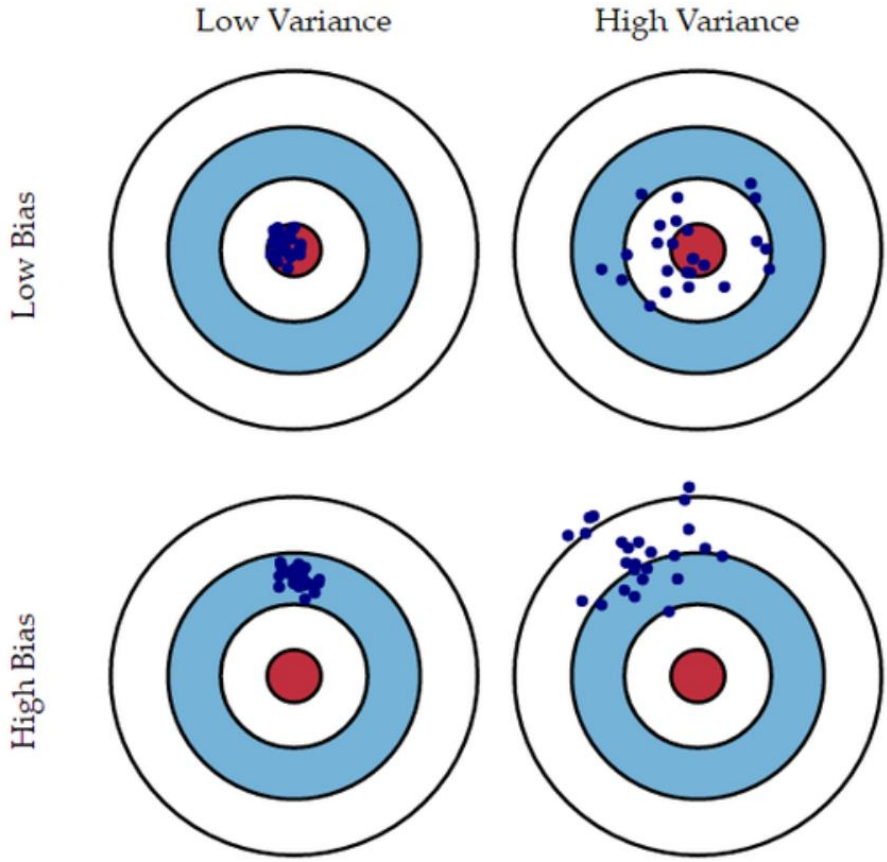


Fig. 6 Image of bias and variance

4.2. Comparison

Let's get started from logistic regression. Obviously, the performance of logistic regression (LR) is lower than random forest (RF) with nearly 20 percent, no matter precision or recall, which means it has a high bias for the dataset. As discussed before, bias commonly results from model being too simple. From formula (1), the LR works based on a linear model, each weight w_i only controls one feature x_i. It performs well dealing with linearly separable dataset. When it comes to linearly inseparable dataset, limited by its own model, LR is not capable to classify dataset perfectly.

It should be noted that linear model doesn't means a line, it exists in different forms in different dimensional space. For instance, it means a line in 2-dimensional space and a plane in 3-dimensional space. A better understanding could be achieved from the Figure 7 below. There are 2 classes here, the green circle represents one while the red cross represents the other one. They are not linearly separable. No matter how many times the parameters are tuned of the model (17), it can't separate them with high accuracy. Obviously, stroke dataset is not linearly separable, that's why LR doesn't perform well.

$$\hat{y} = \sigma(w_0 + w_1 x_1 + w_2 x_2) \tag{17}$$

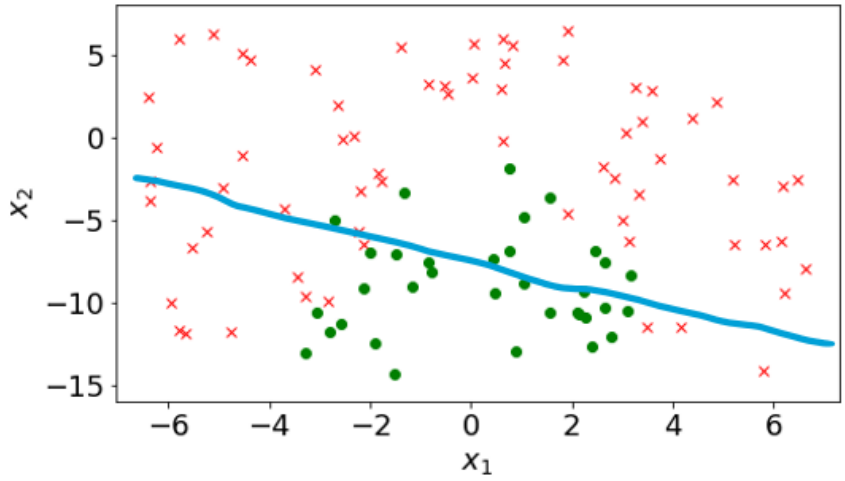


Fig. 7 Image of linear model

When it turns to random forest, it can be seen form its principle that each RF consists of many decision trees. RF make predictions by taking bagging method. However, compared with linear models, models in decision trees are more complex, leading to high variance potentially. How can RF reduce variance and give such a good performance? To understand it, it is needed to get focused on bagging method.

In bagging method, each decision tree gives its prediction independently. Then the predictions are combined and converted through formulas (11) and (12) in the case of binary classification. The formula's function is to calculate the average of all predictions. In general, models averaging is an useful approach to reduce variance. It can be explained in the simple example below. In this example, samples are distributed like Figure 8, 9, and 10. After fitting many models formula (18) to the samples, there is a high variance among models. However, the average model seems the same as the sample space. It works the same way in random forest.

$$\hat{y} = w_0 + w_1 x + w_2 x^2 \ldots + w_9 x^9 \tag{18}$$

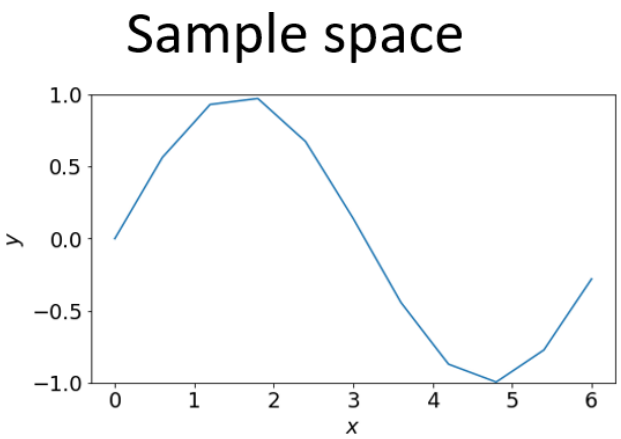


Fig. 8 Image of sample space

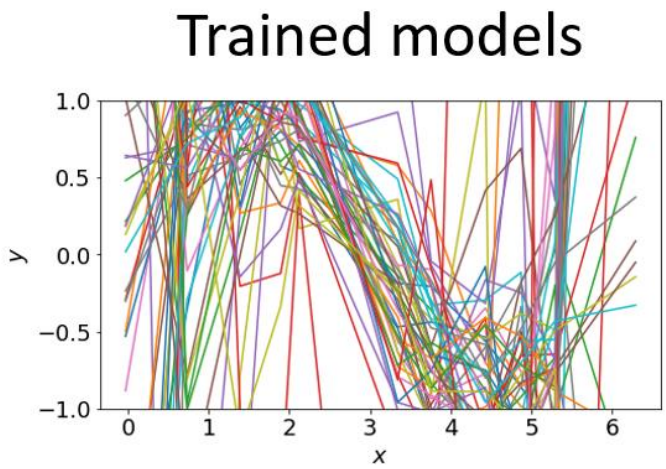


Fig. 9 Image of trained models

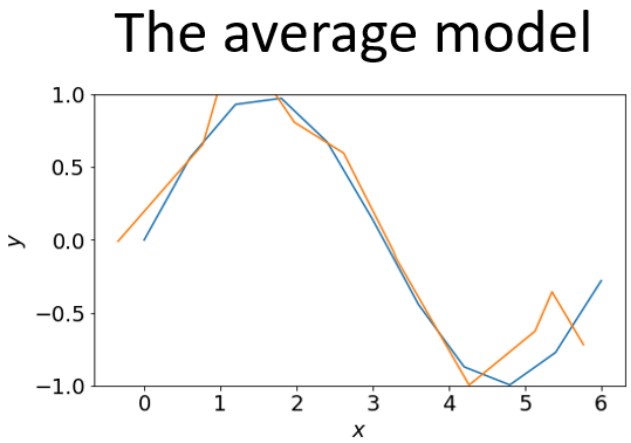


Fig. 10 Image of the average model

4.3. Proposition

As far as concerned, there are still some solutions which can improve models' performance. Firstly, the stroke dataset contains 8000 samples in total, which may not be adequate to train a complex model like random forest. Increasing the number of samples seems a useful approach to promote performance. In addition, label encoding method is used to convert non-numerical features. Compared with one-hot vector, it adds discrepancies to the feature potentially. Therefore, applying other labelling method like one-hot vector may be useful.

5. Conclusion

In this paper, two machine learning method are adopted, logistic regression, and random forest algorithms. After comparing their performance in multiple aspects, there is no doubt that random forest performs better. It gives more ideal results on all evaluation indicators. In discussion part, there are discussions between models' bias and variance, explaining how complexity influences models' performance. It should be noted that making a balance between bias and variance is vital to implement a good model. However, there are still advantages in the study. For example, only two models are adopted here. Implementation of other model like neural networks or support vector machine comes with more comparisons, from which a better understanding could be achieved of the study and improve this study. The author sincerely hope this study can help with stroke prevention and treatment, saving patients' lives from stroke.

References

[1] Wolfe C. The impact of stroke. British medical bulletin, 2000, 56(2): 275-286.

[2] Lo E H, Dalkara T, Moskowitz M A. Mechanisms, challenges and opportunities in stroke. Nature reviews neuroscience, 2003, 4(5): 399-414.

[3] Mozaffarian D, Benjamin E J, Go A S, et al. heart disease and stroke statistics—2016 update: a report from the American Heart Association. circulation, 2016, 133(4): e38-e360.

[4] Boehme A K, Esenwa C, Elkind M S V. Stroke risk factors, genetics, and prevention. Circulation research, 2017, 120(3): 472-495.

[5] Lei Y, Yang B, Jiang X, et al. Applications of machine learning to machine fault diagnosis: A review and roadmap. Mechanical Systems and Signal Processing, 2020, 138: 106587.

[6] Jordan M I, Mitchell T M. Machine learning: Trends, perspectives, and prospects. Science, 2015, 349(6245): 255-260.

[7] Bi Q, Goodman K E, Kaminsky J, et al. What is machine learning? A primer for the epidemiologist. American journal of epidemiology, 2019, 188(12): 2222-2239.

[8] Khosla A, Cao Y, Lin C C Y, et al. An integrated machine learning approach to stroke prediction. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. 2010: 183-192.

[9] Sirsat M S, Fermé E, Câmara J. Machine learning for brain stroke: a review. Journal of Stroke and Cerebrovascular Diseases, 2020, 29(10): 105162.

[10] Fed Soriano, Stroke Prediction Dataset, 2020, URL: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset