Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/46/20241068

Prediction of cardiovascular and cerebrovascular diseases based on machine learning models

Hongji Liu[1*], Yadong Tian[2], Donghong Yu[3]

[1]College of Intelligent Equipment, Shandong University of Science and Technology, Tai 'an 271019, China
[2]Computer science, South Central Minzu University, Wuhan, 430074, China
[3]Engineering, University of Waterloo, University Avenue West, Waterloo, Canada

*corresponding author: 202123030309@sdust.edu.cn

Abstract. Recently we had the fact that cardiovascular disease has become one of the major threats to human life, which leads to the significance of the research around the prevention and cure of such disease. Recently, machine learning algorithms are utilized for the prediction of a certain person who has an illness or not. To verify the effectiveness of predicting cardiovascular disease using machine learning methods, we predict cardiovascular disease given features of a person's life habits and illness history from the Behavioral Risk Factor Surveillance System. Therefore, 5 models are selected, including SVM, logistic regression, decision tree, fully connected network, and XGBoost to evaluate the performance via confusion matrix and ROG curves. Plus, the dataset is highly unbalanced, so we also implemented SMOTETomek re-sampling algorithms to evaluate the models' performance on such kinds of datasets. Results exhibited that XGBoost performs the best on the given dataset, hence deep research on improving the performance using XGBoost is highly recommended.

Keywords: Machine Learning, cardiovascular disease, cerebrovascular disease, prediction

1. Introduction

Artificial intelligence is becoming the force that pushes humanity into the age of intelligence. Today, artificial intelligence is booming, the related technology industry continues to mature, and artificial intelligence is more and more applied in medicine. At the same time, there is a high incidence of cardiovascular and cerebrovascular diseases and a high mortality rate worldwide. In the United States, Heart disease has become the major death of mankind. In the United States, about 695,000 people in 2021 died from heart disease. Heart disease costs about $239.9 billion each year from 2018 to 2019 in the United States [1]. The spending includes the cost of many aspects, like medical fees, health care services, and lost productivity because of death. Cardiovascular disease is difficult to prevent, and the occurrence of sudden situations, the current cardiovascular and cerebrovascular diseases mainly rely on human examination making it difficult to achieve immediate prevention. This situation can be improved through artificial intelligence, and patients can be treated in time. At present, many scholars are committed to this direction. Paper [2] focuses on unsupervised machine learning recall, using k-means algorithms to consider predictive risk factors for heart disease. This project adopts the unbalanced dataset of cardiovascular and cerebrovascular diseases from the Behavioral Risk Factor Surveillance

DOI: 10.54254/2755-2721/46/20241068

System (BRFSS), uses artificial intelligence to predict the incidence of cardiovascular and cerebrovascular diseases, verifies the application of artificial intelligence in this field, and finds ways to train the model.

2. Methods

In this project, we are going to evaluate 5 classical classification algorithms, including Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), XGBoost (XGB), and Fully Connected Neural Network (FCN) on our datasets to determine their performances [3-5]. This section will present a detailed description of those algorithms to better understand the principles' insight. First, data preprocessing should be conducted, followed by the methods we use.

2.1. Data Preprocessing

By analyzing the dataset, we found that the data is highly unbalanced, which shows that the number of negative samples is highly different from the positive samples, which will be shown in the result section. Based on the situation, if a model is directly applied to the dataset, the recall of the minority samples would not be as satisfying as expected due to the high bias of the dataset. Thus, we need some algorithms to balance between the samples to boost the performance of the models. There are multiple methods on dealing with unbalanced data, such as re-sampling by generating new samples, GAN-based re-sample [3,6], and so on. In this project, we decided to use the SMOTETomek algorithm to resample the data. The description of the algorithm is shown below.

The first part of the algorithm is Smote, also known as the Synthetic Minority Oversampling Technique, is an improved random oversampling algorithm. What it has done is to randomly add new calculated minor samples to the dataset to maintain balance. The process of Smote algorithms is shown as follows [7]:

- For a minority sample x, obtain the Euclidean distance between each minority sample and x, obtaining its k nearest neighbors.
- By setting a sampling rate of N based on the imbalance of the dataset, select a few samples based on N and those nearest neighbors of x, and denote one of them x_n.
- Construct a new sample based on the following formulation.

$$x_{new} = x + rand(0,1)\left|x - x_n\right| \tag{1}$$

By the algorithms above, new samples of the minority class were generated to balance the dataset. However, this has a problem in that the generated data is uncontrollable due to the setting of the sampling rate, which is not adjustable with ease. Thus, a down-sampling algorithm is proposed.

The second part of SMOTETomek is the Tomek algorithm, which down-samples the majority class of the dataset. The algorithm is proposed to find a pair of samples, (a, b), which are called TomekLinks. A TomekLinks is defined as a pair of samples that not only belong to different classes but also are KNN mutually. By deleting the TomekLinks of the dataset, the majority class will be down-sampled and close to the minority class. Figure 1 shows how the TomekLinks was defined and balanced.

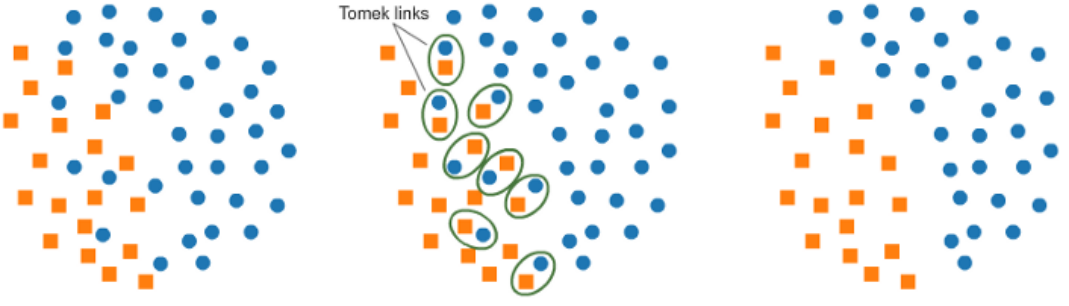


Figure 1. Finding TomekLinks and deleting to balance the data.

DOI: 10.54254/2755-2721/46/20241068

By combining Smote and TomekLinks, we have the SMOTETomek algorithm, in which we use Smote to generate data, and TomekLink to downsample, which would be able to balance the sample classes and improve the performance of the predicting model.

2.2. Models Description

- Logistic Regression (LR)

LR is derived from linear regression, which is one of the most common generalized linear algorithms in machine learning. The algorithm could be explained as the following formulation [6]:

$$y = f(\boldsymbol{w}^T x + b) \tag{2}$$

While $f(x)$ is a sigmoid function, which is expressed as:

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$

Combining the equation above, we get:

$$P(y \mid x; \boldsymbol{w}) = g(\boldsymbol{w}^T x + b) = \frac{1}{1+e^{-(\boldsymbol{w}^T x+b)}} \tag{4}$$

Denote that $P(y|x; \boldsymbol{w})$is the predicted probability of a given feature vector x, while y is the predicted label either positive or negative. W is the weighted feature defined by the input vector, and b is the bias, while both w and b are trainable. The output of linear regression is limited within a line approximately depending on the input and limiting the output to whether 0 or 1 is not feasible. Logistic regression brings in the sigmoid function to increase the non-linearity of the algorithms, which transfers the output of the linear regression to the output probability. By setting the output threshold as 0.5 of the sigmoid function, the model will predict a label as positive with a given feature while $(\boldsymbol{w}^T x + b)$ is greater than the threshold and negative while smaller than the threshold.

- Support Vector Machine (SVM)

SVM is another classical binary classification algorithm that has a more robust ability to solve the binary classification problem by finding a hyperplane to divide the samples into 2 groups with some specific conditions to be met. Its characteristics determined its advantages in sample space that are linearly separable. Based on the description in [6], Let's say we have a dataset that represents as $D = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$, where $y^{(n)} \in \{+1, -1\}$. If D is linearly separable, it should exist a specific hyperplane:

$$\boldsymbol{w}^T \boldsymbol{x} + b = 0 \tag{5}$$

Which can linearly separate 2 different classes, then for each sample, we have:

$$y^{(n)}(\boldsymbol{w}^T \boldsymbol{x}^{(n)} + b) > 0 \tag{6}$$

The distance between any samples and the hyperplane $\gamma^{(n)}$ is defined as:

$$\gamma^{(n)} = \frac{\left|\boldsymbol{w}^T \boldsymbol{x}^{(n)} + b\right|}{\|\boldsymbol{w}\|} = \frac{y^{(n)}(\boldsymbol{w}^T \boldsymbol{x}^{(n)} + b)}{\|\boldsymbol{w}\|} \tag{7}$$

DOI: 10.54254/2755-2721/46/20241068

Figure 2 shows how the SVM works. We can see that the greater the is, the more robust for SVM. Usually, SVM can utilize the core function to map the samples into higher dimensional space to solve the problem that samples are not linearly separated in the current dimensional space.

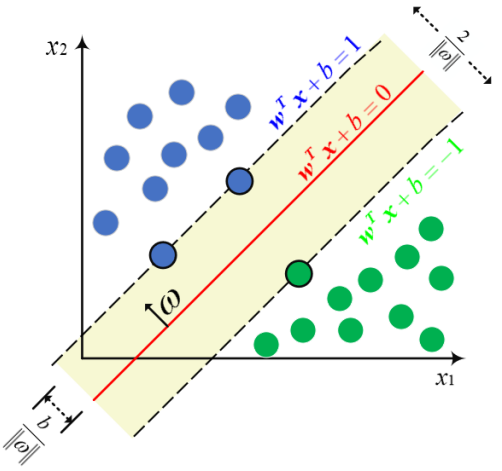


Figure 2. Principle of Support Vector Machine

- Decision Tree (DT)

DT is one of the most common tree algorithms to be utilized in classification, which similar to normal tree structures, has a root node, branches, and leaf nodes. Figure 3 shows a very basic decision tree based on what types of games a person likes to play.

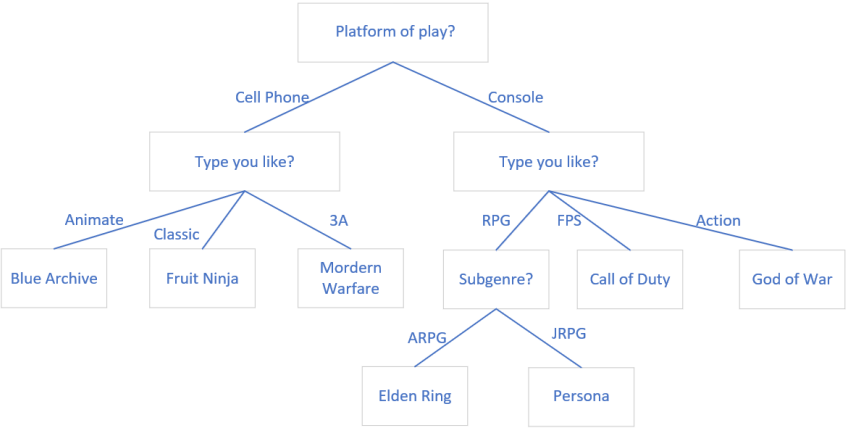


Figure 3. A decision tree example based on gaming recommendation.

Judging from the structure, clarity is one of the biggest advantages of a decision tree, which does not need the data to be pre-processed, thus maintaining intact information of the dataset and explanation of the algorithm [4].

Some algorithms assist in building a tree, including ID3, C4.5, and CART, while the splitting rules could depend on information gain, Gini index as well and log-loss [4,8,9]. In this project, we build a decision tree based on information gain G(X, Y), which is defined as follows:

$$G(X,Y) = H(Y) - H(Y \mid X) \tag{8}$$

Here H(Y) is the information entropy of a given event Y [9]:

DOI: 10.54254/2755-2721/46/20241068

$$H(Y) = -\sum_{i=1}^{N} P(y_i) \log p(y_i) \tag{9}$$

It represents the uncertainty of whether Y is happening or not. H(Y|X) is the conditional entropy, which represents the entropy of Y given condition X. It can be defined as follows:

$$\begin{aligned} H(Y \mid X) &= \sum_{x \in X} P(x) H(Y) \\ &= -\sum_{x \in X} P(x) \sum_{y \in Y} P(y \mid x) \log P(y \mid x) \\ &= -\sum_{x \in X} \sum_{y \in Y} P(x, y) \log P(y \mid x) \end{aligned} \tag{10}$$

By subtracting the conditional entropy of Y, the information gain shows that using another variable X that affects Y, the uncertainty of Y decreased. The training of the decision tree is to find that selected feature to maximize the information gain of the given feature space, which makes the classification more feasible.

- Neural Network (NN)

NN is one of the most commonly used artificial intelligence technologies that used in various areas. The basic element of the neural network is a neuron in each layer. Figure 4 shows how a neuron works.

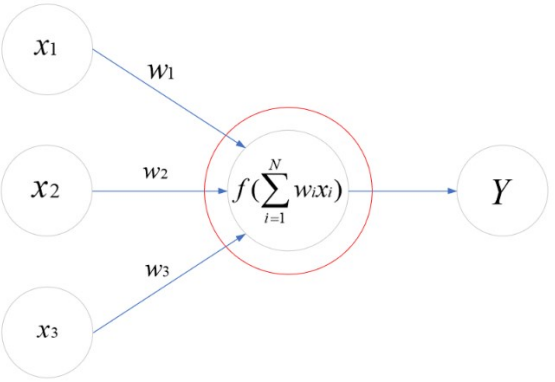


Figure 4. A basic structure of a neural network

The output Y is formulated as follows:

$$Y = f(\sum_{i=1}^{N} w_i x_i) \tag{11}$$

The red circle in Figure 4 represents a neuron in a layer. By accepting a weighted sum from the input or output of the previous layers, the neuron will output to the next layer by an activation function, which increases the nonlinearity of the network and increases the learning ability to adapt more complex features of the input. Multiple neurons are stacked to construct a layer, and multiple layers are connected to become NN.

Because the dataset is represented by a form, we use a fully connected neural network for evaluation. Figure 5 shows the structure of a fully connected neural network (FCN).

DOI: 10.54254/2755-2721/46/20241068

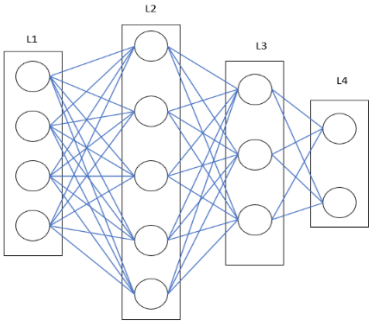


Figure 5. A fully connected network.

Figure 5 shows that each neuron is connected in different layers. L1 is called the input layer, L2 and L3 is called the hidden layer, which computes the information given by the input layer, and L4 is called the output layer, which outputs the result of the neural network, labels for classification, and numbers for regression. In this project, we use a 5-layer FCN, with the numbers of neurons sequentially, 64, 128,32,20,1, while the last is the output layer.

- XGBoost

Proposed and summarized by [8], XGBoost is an upgraded version of gradient boosting (GBDT), with multiple improvements compared to its predecessor. Paper [5] tells us that, unlike a sole decision tree, gradient boosting linearly adds up the results from multiple weak classifiers, which is multiple decision trees, and updates the next group of weak classifiers by residuals from the previous classifiers, which is the distance between predicted value and label. The object function of XGBoost could be written as the following equation.

$$\begin{aligned} Obj^{(t)} &= \sum_{i=1}^{n} l(y_i, y\wedge_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_t) \\ &= \sum_{i=1}^{n} l(y_i, y\wedge_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + c \end{aligned} \tag{12}$$

The sub-function is the cost function of XGBoost and is the regularization term. $y\wedge_i^{(t)}$ is expressed as $y\wedge_i^{(t)} = y\wedge_i^{(t-1)} + f_t(x_i)$, where is the current model, $y\wedge_i^{(t-1)}$ is the previous model that finished training in the previous steps, and is the training model of the current step. The object is to find the that minimizes the object function. According to [8], we can use the Taylor expression, which is defined as:

$$f(x+\Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2 \tag{13}$$

Thus, by expanding f_t using Taylor's expression, the objection function could be written as:

$$Obj^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, y\wedge_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + c \tag{14}$$

$$g_i = \partial_{y\wedge^{(t-1)}} l(y_i, y\wedge^{(t-1)}) \tag{15}$$

$$h_i = \partial^2_{y\wedge^{(t-1)}} l(y_i, y\wedge^{(t-1)}) \tag{16}$$

DOI: 10.54254/2755-2721/46/20241068

Unlike GBDT, XGBoost uses the second-order derivates in the general objection function not only to increase the possibilities of self-define objection functions to improve generalizability but also to speed up while doing gradient descent, which describes the direction of changing gradients.

Paper [8] also mentions that XGBoost has a regularization term, which is in the objection function. $\Omega(f_t)$ depends on two parts, including the number of leaves T, and the L2 norm of the score of each tree. The regularization term is defined as follows:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{17}$$

The γ and λ are both hyperparameters and need to be adjusted by human beings. With this regularization term, XGBoost can self-adjust each sub-tree and prevent overfitting, which increases its stability and robustness in predicting.

In addition to the previous improvements, XGBoost also has the ability of parallel computing to speed up training, proper methods of dealing with null values, and so on, which makes it one of the most welcome machine learning algorithms in research and industrial aspects.

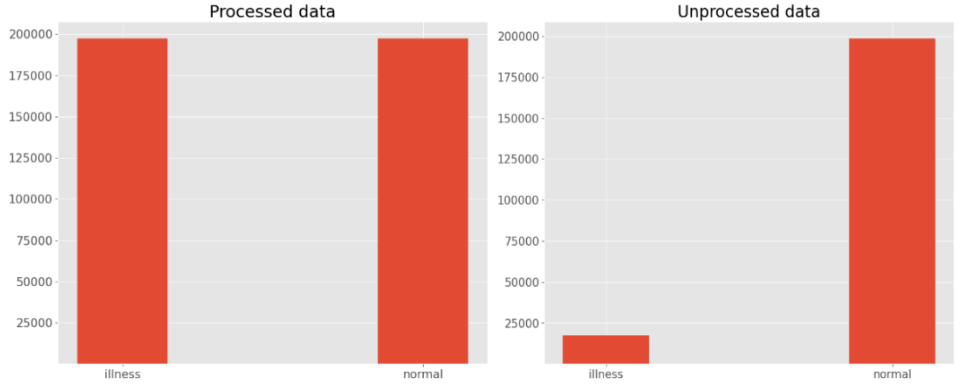
3. Result

3.1. Experiment Setup

This section will show and discuss the results of our experiment. The dataset we use comes from BFRSS, which contains a group of features of a person's daily life and history of illness to determine if he has a heart attack or not. We split the train and test dataset with 70% for training and 30% for testing. First, we examine the effect of utilizing the data preprocessing algorithm on the dataset. To maintain the confidence of our result, all the models were trained using default parameters, except the neural network, in which each neuron per layer was set to 64, 128,32,20,1 sequentially from input to output. We then analyze the model performance using the confusion matrix and ROC curve and discuss the results we got.

3.2. Data Preprocessing Analysis

The obtained data depicts two experimental stages: one conducted without the preprocessing using the SMOTETOMEK algorithm, and the other after applying the said algorithm to balance the dataset. The figure 6 shows that the unprocessed data has highly unbalanced characteristics, with negative labels about 20k and positive labels lower than 2.5k. By applying the SMOTETomek algorithm, the 2 classes of labels are nearly balanced, which is a desired attribute of the dataset that is sent to the model. This shows the effectiveness of having data re-sampling techniques to the dataset to clean the data and is a significant part of improving the performance of the model.



(a) (b)

Figure 6. The visualization of 2 labels in the dataset with (left) and without (right) SMOTETomek.

DOI: 10.54254/2755-2721/46/20241068

3.3. *Model Analysis*

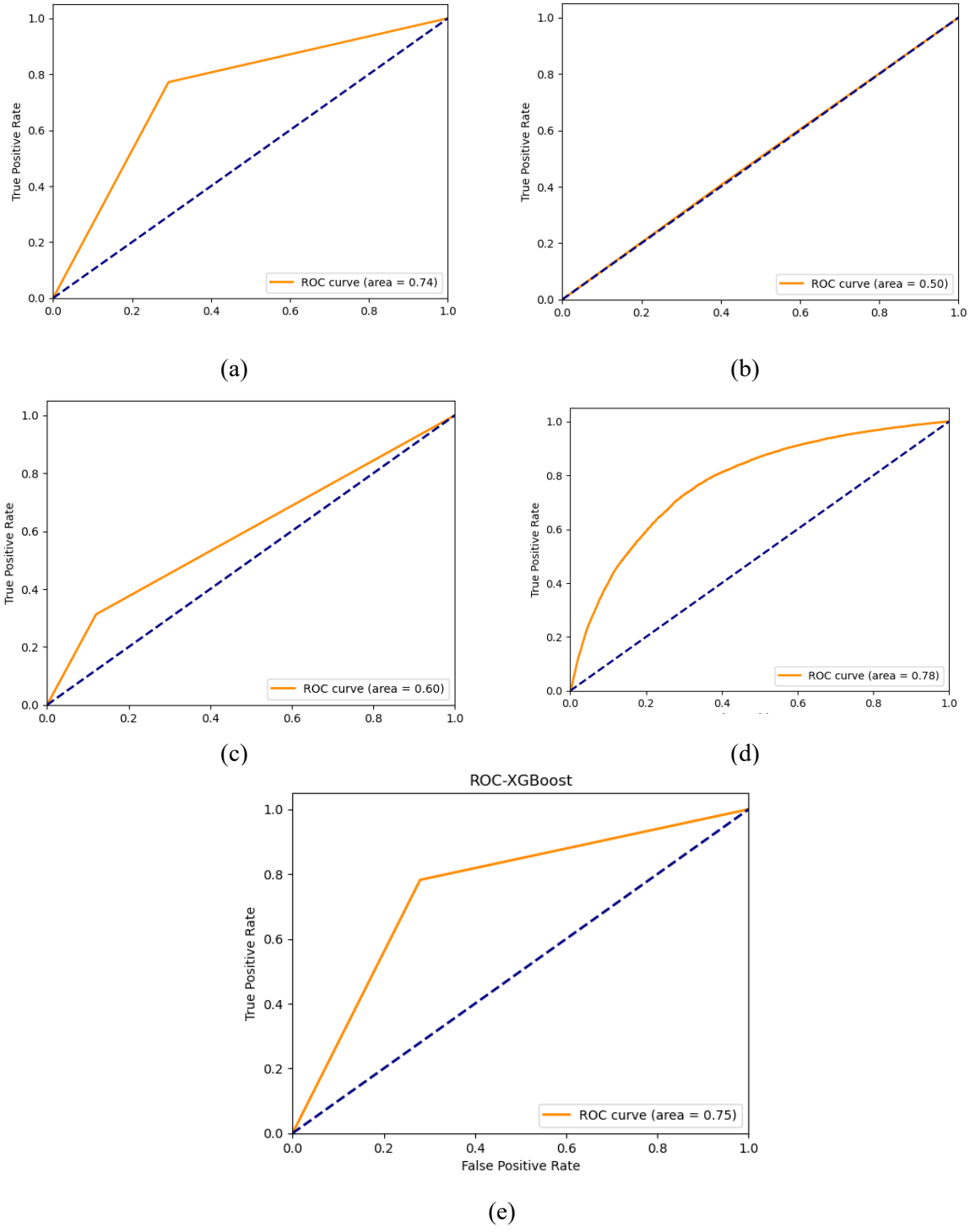


Figure 7. (a) to (e) depict the ROC curve sequentially from Logistic Regression, Support Vector Machine, Decision Tree, Neural Network, and XGBoost

DOI: 10.54254/2755-2721/46/20241068

Table 1. The performance of 5 models

Method	Precision	Recall	F1-score	Accuracy
Logistic Regression	0.58	0.74	0.56	0.71
Support Vector Machine	0.50	0.50	0.43	0.58
Decision Tree	0.56	0.60	0.57	0.83
Neural Network	0.58	0.70	0.58	0.77
XGBOOST	0.59	0.75	0.57	0.73

We evaluate 5 models using the metrics of the ROC curve and converted confusion matrix index, which is the precision, recall, and F-1 score, with the average accuracy as a reference. Figure 7 depicts the ROC curve of 5 models and Table 1 shows the quantity results of the performance from 5 models. Based on the results, we have the following findings:

1. The ROC curve shows that the model we choose has satisfactory performance, except SVM, which has nearly no differences compared to the reference line, which is y = x. This shows that SVM cannot nearly determine whether a person is suffering from heart disease or not.
2. The table shows that the precision of each model is nearly the same, ranging around 0.5 to 0.6. However, recall shows that logistic regression and XGBoost are comparatively better than the others, followed by FCN with about 0.05% lower.
3. The F1-score of the 5 models shows that except SVM, the rest of the models are having approximately the same.

The results shows that XGBoost, logistic regression and neural network are having satisfying results on the prediction. However, we usually choose the maximum of the recall and F1-score in order to predict if someone is illness or not, which is more significant in predicting illness. Considering the training efficiency and overall metrics, we conclude that XGBoost performs the best among 5 models, which means that using XGBoost to predict illness on unbalanced data is the one of the best choices on machine learning algorithms.

3.4. Discussion

In the comparative experimentation involving LR, SVM, DT, FCN and XGBoost, the XGBoost model manifests several conspicuous advantages. It preserves a discernible equilibrium amid precision and recall metrics, particularly excelling in the treatment of imbalanced and extensive datasets. The XGBoost model exhibits steadfast performance in terms of the F1-score metric, underscoring its suitability for diverse contexts. Its ensemble learning characteristics confer elevated predictive accuracy. To synthesize, within the experimental milieu, XGBoost emerges as competitively advantageous in the realm of classification challenges. However, it is imperative to recognize that while XGBoost demonstrates these advantages within the outlined data contexts, such advantages do not automatically translate to it being the optimal choice for all situations. The selection of a model must still consider the specific problem domain, unique data characteristics, and other pertinent factors.

4. Conclusion

In this project, we evaluate 5 machine learning models on an unbalanced dataset of predicting a person if he has potential cardiovascular disease given features described his personal life and illness history. The result shows that except SVM, all of models shows the ability to complete the task of predicting illness given an unbalanced data. Among them were the XGBoost which outperformed the rest. Besides, this project has demonstrated the effectiveness of the SMOTET to make algorithm for dealing with imbalanced data, providing ideas for further exploration of the processing of imbalanced datasets in the future. However, in this experiment, it was found that there are still certain problems. The data dimension is too large, and in the future, feature engineering or dimensionality reduction algorithms will be carried out to further improve the predictive ability of the model. The confusion matrix shows a high

DOI: 10.54254/2755-2721/46/20241068

FP, which affects the overall accuracy. In the future, the algorithm will be adjusted according to this situation to solve this problem.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Indrakumari R, Poongodi T, Jena S R. Heart disease prediction using exploratory data analysis[J]. *Procedia Computer Science*, 2020, 173: 130-139.

[2] Heart disease facts Centers for Disease Control and Prevention. 2003: https://www.cdc.gov/heart isease/facts.htm.

[3] Engelmann J, Lessmann S. Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications,* 2021, 174: 114582.

[4] Patel, H.H. and Prajapati, P. Study and analysis of decision tree-based classification algorithms', *International Journal of Computer Sciences and Engineering,* 2018 **6(10).** 74–78.

[5] He Z, Lin D, Lau T, et al. Gradient boosting machine: a survey. *arXiv preprint arXiv:1908.06951*, 2019.

[6] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. *Journal of artificial intelligence research,* 2002, **16**: 321-357.

[7] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C]//*Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.* 2016: 785-794.

[8] Zhihua Z 2016, Machine Learning, *Tsinghua University Press*.